

# Webs/blogs con Quarto

(Web del curso [aquí](#))

26 de junio de 2023

# Intro

---

- En el [I Congreso & XII Jornadas de Usuarios de R](#) celebradas en Córdoba en noviembre de 2022 impartí el taller [Mi primer blog con Quarto](#)
- Buena parte de estas slides provienen del taller, así que algunas se parecen un poco 😊

# Estructura del ... 🤗 “taller”

0. **Primeras ideas**, visionado de blogs
1. Creación de un blog básico: mi blog en 3 minutos
2. **Tuneado básico** del blog.
3. Convirtiendo el blog en una **web personal**
4. Workflow: **creando posts**
5. **Escribiendo** un post con Quarto
6. Tuneado: **mejorando la web**
7. Gestión y **publicación** de la web/blog

# Objetivo(s) del taller

- Aprender a hacer websites/blogs con Quarto
- Aprender sobre Quarto mientras vamos creando una web/blog
- Que salgáis todos con el bosquejo y la **intención de hacer** una web/blog!!
- ... y, si puede ser, **pasar un buen rato** 😊 mientras aprendemos

# Beneficios de tener un blog

- Se aprende mucho!!
- Es **fácil** (pocas barreras de entrada si ya usas Rmarkdown)
- Es **divertido**
- Registrar análisis que haces que de otra forma olvidarías
- Portafolio y contactos
- Productive procrastination (!!)

(Discursos más elaborados [aquí](#) o [aquí](#) )

## Importante:

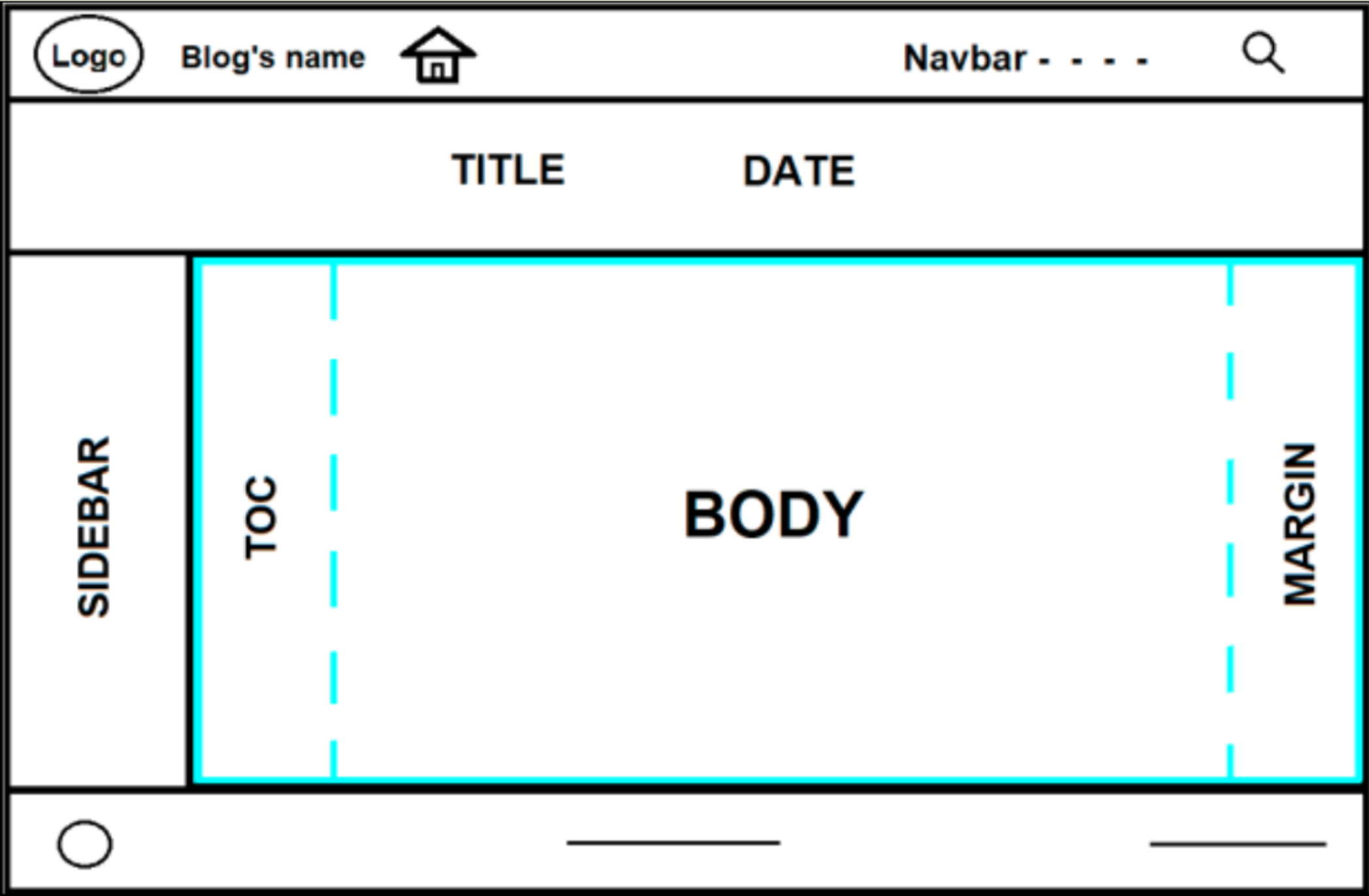
- El blog **no tiene que ser perfecto**. Mejor empezar con un blog sencillo
- Los posts **tampoco tienen por qué estar perfectos** ... y [Jenny Bryan está de acuerdo conmigo](#)

# Primeras ideas

---

- Vamos a usar Quarto para **hacer un blog**
- En realidad un blog no es más que un tipo (especial) de web

# Estructura de una web



HEADER/Navbar

TITLE-block banner

ARTICLE

Footer

# Ejemplos de blogs (hechos con Quarto)

- [Mike Mahoney](#) (el repo [aqui](#))
- [Isabella Velásquez](#) (el repo [aqui](#))
- [Danielle Navarro](#) (el repo [aqui](#))
- [Jeff y Marc Dotson](#) (el repo [aqui](#))
- [Layton R blog](#) (el repo [aqui](#))
- [R Lille](#) (el repo [aqui](#))
- [Drew Dimmery](#)
- [PA blog](#) (el repo [aqui](#))
- [Mario Angst](#)
- [Matt Worthington](#) (el repo [aqui](#))
- [Ilya Kashnitsky](#) (el repo [aquí](#))
- [Rebecca Carter](#) (el repo [aquí](#))

## Más opciones:

- La [Gallery de Quarto](#)



# Ejemplos de webs (hechas con Quarto)

- [Making shareable documents with Quarto](#) y el [repo](#)
- [Data in life](#) y el [repo](#)
- [Andrew Heiss](#) y el [repo](#)
- [Program Evaluation for Public Service](#) y el [repo](#)
- [Comparative Public Administration](#) y el [repo](#)
- [Quantum Jitter](#) y el [repo](#)
- [Ella Kaye](#) y el [repo](#)
- [Introduction to Regression and Data Science](#) y el [repo](#)
- [STA 199: Intro to Data Science](#) y el [repo](#)
- [ESPM 157: Data Science for Global Change Ecology](#) y el [repo](#)
- [MAT555E-Statistical Data Analysis for Comp. Sciences](#) y el [repo](#)
- [Antony Barja](#) y el [repo](#). Además tiene cursos sobre GIS
- [The tidy trekker](#) y el [repo](#)
- [Real World Data Science](#) y el [repo](#)
- [Tidy Tales](#) y el [repo](#)

- Unas [plantillas](#) para webs [personales](#) y [académicas](#) (cursos y workshops)
- Otras [plantillas](#) para webs académicas
- Más [plantillas](#), pero también para books, etc ...**Muy chulas**
- Otras [plantillas](#) para webs, pero también para books, etc ... (muy chulas)

# 1. Creación de un blog básico con Quarto

(1ª parte del tutorial)

## ¿Qué haremos en este tutorial?

- Veremos de forma rápida el **proceso de creación** y la **estructura de ficheros** de un blog creado con Quarto. La documentación oficial está [aquí](#).
- Alojaremos el blog en [Quarto Pub](#).
- Será un **blog básico pero completamente operativo**. Ya tendremos tiempo de mejorarlo, o tunearlo, en las siguientes secciones del taller.

## Tarea 1.1: Hemos de crear la carpeta que contendrá nuestro blog.

La carpeta la creará Quarto, pero tenemos 3 formas de hacerlo:

### A) Crear la carpeta (o Qproject) desde RStudio

- En RStudio sigue la siguiente ruta de menús:

`File > New Project > New Directory > Quarto Blog`

- Mejor si llamas a la carpeta: `blog_prueba_00`

## Extensión a la tarea 1.1: Se puede crear la carpeta de otras formas

### B) Crear la carpeta desde la Terminal

- Has de ejecutar desde la Terminal:

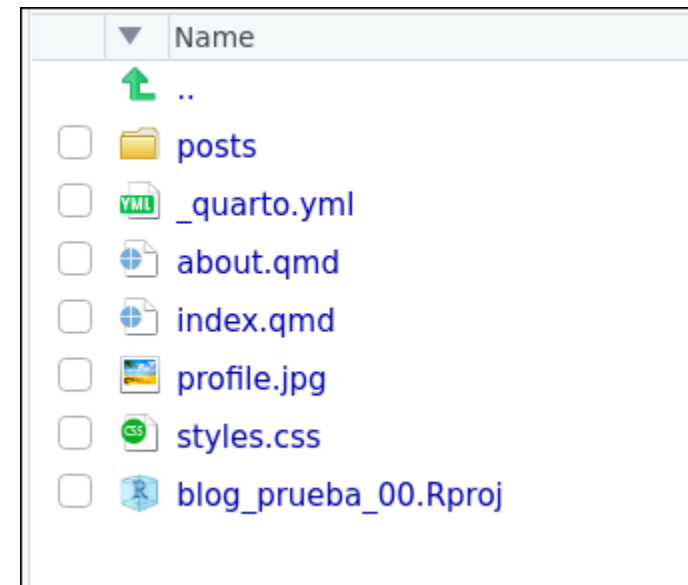
```
quarto create-project blog_prueba_00 --type website --template blog
```

### C) Usando un Qproject ya creado con RStudio Cloud

- Tienes que ir [aquí](#). Se abrirá Rstudio en el navegador con un Qproject llamado `blog_prueba_00`.
- Has de tener cuenta en [RStudio Cloud](#). Haz una copia permanente y trabaja sobre ella

De cualquiera de las 3 maneras **acabaremos teniendo una carpeta con todo o necesario para crear un blog.**

# Contenido del Qproject



Los archivos importantes ahora son 3:

- `_quarto.yml`: especifica la estructura (y apariencia) de la web
- `index.qmd`: generará (y dará formato) a la “landing page” o Home. Esta página será un listado de páginas (un blog)
- `about.qmd`: una de la páginas del blog

## Tarea 1.2: Generación del blog

- Ir al cuadrante superior-derecha de RStudio, pinchar en la pestaña **Build** y pinchar en **Render Website**.
- De esta forma, se procesarán los archivos del Qproject y se creará el blog.
- Podremos verlo en el cuadrante inferior-derecha de RStudio, concretamente en la pestaña **Viewer**.

## Extensiones: formas alternativas de generar el blog

- También podríamos haber generado el blog usando la Consola o la **Terminal** desde RStudio.
- En ambos casos hay que estar en el Qproject “blog\_prueba\_00”; es decir, “blog\_prueba\_00” es el working directory

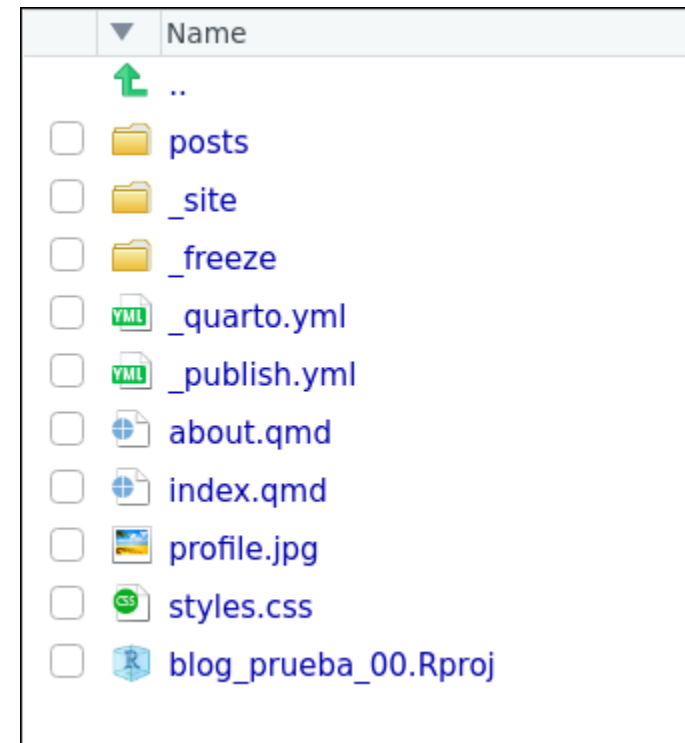
- Desde la **Terminal**

```
quarto render
```

- desde la **Consola de R**

```
install.packages("quarto")  
quarto::quarto_render("index.qmd")
```

# Contenido del Qproject (tras procesar el blog)



- Se han creado 2 subcarpetas: `_site` y `_freeze`
- `_site` es la carpeta que contiene el blog.



# Ver el blog en local

- `_site` es la carpeta que contiene el blog. Veamos su contenido
- La página principal (o Home) de nuestro blog es el archivo `index.html`.
- El archivo `index.html` ha sido generado por el fichero `index.qmd`.

## Tarea 1.3: Ver el blog

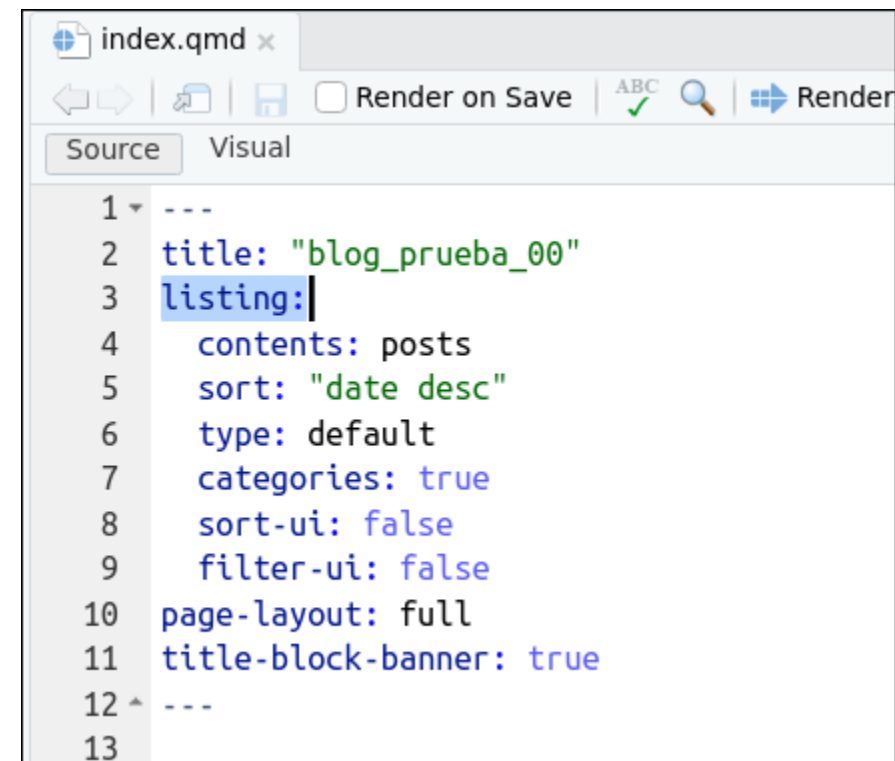
- Enseguida publicaremos el blog en Quarto, pero antes vamos a verlo en local.
- Para ello solo tienes que pinchar en el archivo `./blog_prueba_00/_site/index.html`
- Ese fichero es el que contiene la “Landing Page” o “Home” de nuestro blog.
- Como ves el blog es muy sencillo, solo tiene 2 páginas:
  - la Home Page (que consiste en un listado de posts)
  - la página `About`

# Contenido de `index.qmd`

- El archivo `index.qmd` genera `index.html`, la página principal (o Home) de nuestro blog

Extensiones: ver el contenido del archivo `index.qmd`

Contenido del archivo `index.qmd` que genera a `index.html`:

A screenshot of a code editor window titled 'index.qmd'. The editor has a toolbar with icons for navigation, saving, and rendering. Below the toolbar are two tabs: 'Source' and 'Visual'. The 'Source' tab is active, showing a YAML-like configuration for a Quarto document. The code is as follows:

```
1 ---
2 title: "blog_prueba_00"
3 listing:
4   contents: posts
5   sort: "date desc"
6   type: default
7   categories: true
8   sort-ui: false
9   filter-ui: false
10 page-layout: full
11 title-block-banner: true
12 ---
13
```

- Como vemos, el archivo `index.qmd` no tiene contenido, sólo contiene un `yaml` o encabezamiento
- Ese encabezamiento genera una lista (`listing:`) con los contenidos (con los ficheros `.qmd`) que haya en la carpeta `posts`.
- Sí, has imaginado bien, en la carpeta `posts` es donde estarán los ficheros `.qmd` que generarán los posts
- El resto del `yaml` son opciones para nuestro blog. Por ejemplo, `sort: "date desc"` ordenará el listado de post de más reciente a más antiguo.
- Lo iremos viendo

# Alojando el blog

## Tarea 1.4: Alojamiento del blog en Quarto Pub

Para publicar nuestro blog en Quarto Pub sólo hay que:

- teclear en la **Terminal** de RStudio (panel inferior-izquierda) lo siguiente:

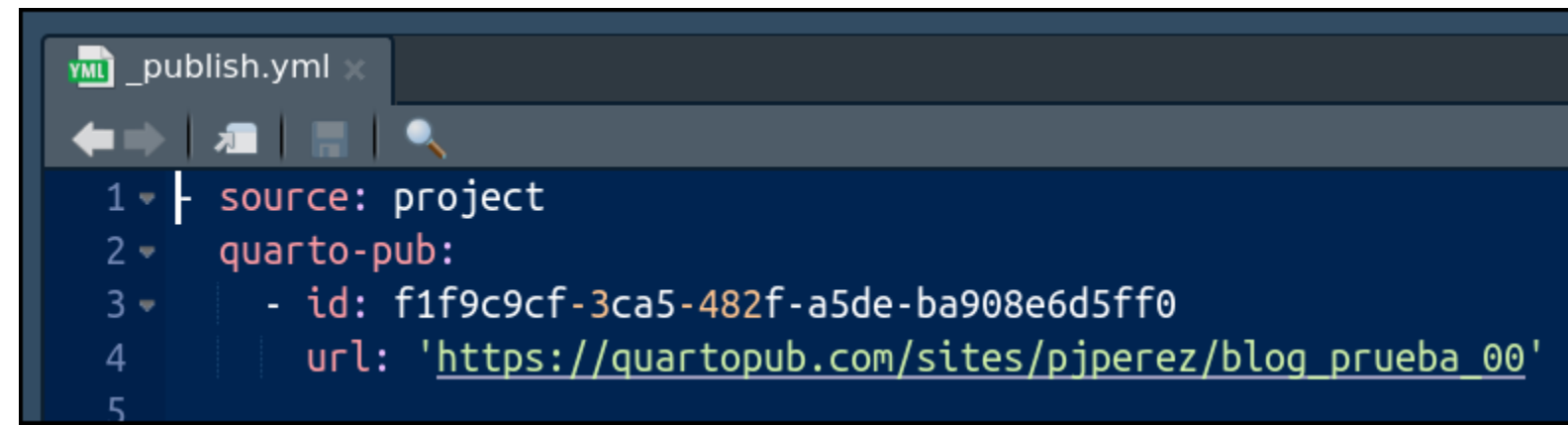
```
quarto publish quarto-pub
```

- Tras ejecutar lo anterior, se te pedirá que autorices en la Terminal.
- Después se abrirá el navegador, tendrás que logearte en [Quarto Pub](#) y volverás a autorizar. Además te preguntarán el nombre del blog.

- Puedes ver [aquí](#), como quedaría el blog una vez alojado en Internet.

## Extensiones: el fichero `_publish.yml`

Si te fijas, tras haberse publicado el blog en “Quarto Pub”, se habrá creado un **nuevo fichero** en nuestro Qproject: es el fichero `_publish.yml`. En él solamente están los metadatos necesarios para publicar y **actualizar nuestro blog**.



```
1 | source: project
2 | quarto-pub:
3 |   - id: f1f9c9cf-3ca5-482f-a5de-ba908e6d5ff0
4 |     url: 'https://quartopub.com/sites/pjperez/blog_prueba_00'
5 |
```

## Resumen (tutorial nº 1)

En este tutorial, la **segunda parte del taller**, hemos visto:

- Como crear un Qproject con la estructura de carpetas y archivos necesaria para generar un blog básico
- Hemos visto (de forma rápida) el contenido, los archivos, que hay en el Qproject.
- El Qproject contiene 3 archivos que serán los importantes, los que trabajaremos en el próximo tutorial, concretamente: `_quarto.yml`, `index.qmd` y `about.qmd`
- Hemos aprendido a generar el blog a través de RStudio, simplemente **Build > Render Website**
- Hemos visto que el blog realmente se aloja en la carpeta `_site` y que la homepage es `index.html`
- La homepage del blog es un listado de post (lógicamente, es un blog!!)
- La homepage (`index.html`) es generada por el archivo `index.qmd`
- El archivo `index.qmd` solo tiene YAML
- Hemos publicado el blog en Quarto Pub

# 2. Tuneado básico del blog

(2ª parte del tutorial)

## ¿Qué haremos en este tutorial?

- En esta segunda parte del taller vamos a **mejorar/tunear un poco nuestro blog**. Nos servirá para afianzar nuestro conocimiento de la estructura de ficheros del blog
- En este tutorial nº 2, para hacer un **tuneado básico** del blog, modificaremos 3 archivos, concretamente los archivos: `_quarto.yml`, `index.qmd` y `about.qmd`. Iremos archivo por archivo
- Finalmente modificaremos (un poco) el archivo `styles.css`
- En el tutorial nº 6 se profundiza en el tuneado del blog

# Archivo `_quarto.yml`

```
_quarto.yml
1 project:
2   type: website
3
4 website:
5   title: "blog_prueba_00"
6   navbar:
7     right:
8       - about.qmd
9       - icon: github
10         href: https://github.com/
11       - icon: twitter
12         href: https://twitter.com
13 format:
14   html:
15     theme: cosmo
16     css: styles.css
17
```

Contenido original de `_quarto.yml`

```
_quarto.yml
1 project:
2   type: website
3
4 website:
5   title: "Mi blog (aún en pruebas)"
6   navbar:
7     logo: "profile.jpg"
8     right:
9       - text: "About me"
10         href: about.qmd
11       - icon: github
12         href: https://github.com/<tu-usuario-github>
13       - icon: twitter
14         href: https://twitter.com/<tu-usuario-twitter>
15       - icon: envelope
16         url: "mailto:<tu-mail@uv.es>"
17     left:
18       - text: "Docencia"
19         href: docencia.qmd
20
21 format:
22   html:
23     theme: flatly
24     css: styles.css
25     toc: true
26     highlight-style: a11y

```

Así dejaremos el fichero `_quarto.yml`



## Tarea 2.1: Modificar el contenido de `_quarto.yml`

- Sustituye el contenido original del fichero `_quarto.yml` por el contenido del siguiente chunk.
- Pon un título que te guste a tu blog (línea 5 `title: "titulo-de-mi-blog"`)
- Pon tu usuario de Github y cuenta de Twitter (líneas 12 y 14: `href:`)
- Pon tu email (línea 16: `href:`)

```
1 project:
2   type: website
3
4 website:
5   title: "Mi blog (aún en pruebas)"
6   navbar:
7     logo: "profile.jpg"
8     right:
9       - text: "About me"
10        href: about.qmd
11       - icon: github
12        href: https://github.com/<tu-usuario-github>
13       - icon: twitter
14        href: https://twitter.com/<tu-usuario-twitter>
15       - icon: envelope
16        url: "mailto:<tu-mail@uv.es>"
17     left:
18       - text: "Docencia"
19        href: docencia.qmd
20
21 format:
22   html:
23     theme: flatly
24     css: styles.css
25     toc: true
26     highlight-style: all
```

Además:

- **líneas 17-19:** añadimos la página/pestaña “Docencia” a la izquierda de la navbar, pero estamos referenciando un archivo que aún no existe, el archivo `docencia.qmd`
- **líneas 21-26:** modificamos el theme, ponemos TOC a los documentos, ..., **CSS**,

## Tarea 2.2: Crear el archivo `docencia.qmd`

- Podemos crear `docencia.qmd` con cualquier contenido, incluso podría ser un documento sin contenido, PERO, mejor si lo creas así:

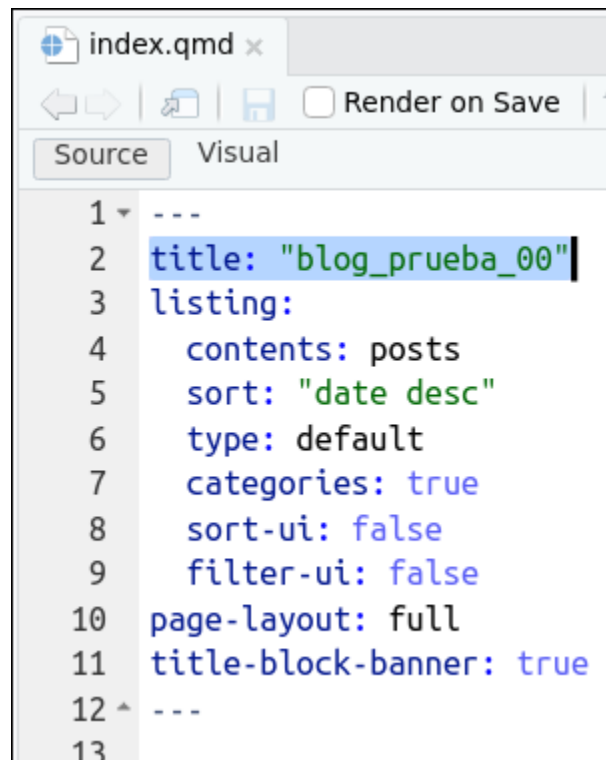
```
1 ---
2 title: "Mi docencia"
3 ---
4
5 Mi docencia es:
6
7 ## curso 2021-2022
8
9 - [Econometría] (./pdfs/econometria.pdf)
10
11 ## curso 2022-2023
12
13 - [Algebra] (/pdfs/econometria.pdf)
14
15 ## curso 2023-2024
16
17 - [Intro a R] (pdfs/econometria.pdf)
```

## Tarea 2.3: Crear el archivo `/pdfs/econometria.pdf`

- Como puedes ver, en el archivo `docencia.qmd` se hace referencia al archivo `/pdfs/econometria.pdf`. Si quieres que el link funcione, tienes que crear dentro del Qproject, la subcarpeta `pdfs`, y dentro de ella el archivo `econometria.pdf`.
- Puede ser cualquier pdf, siempre que se llame `econometria.pdf`

## Tarea 2.4: Volver a generar el blog

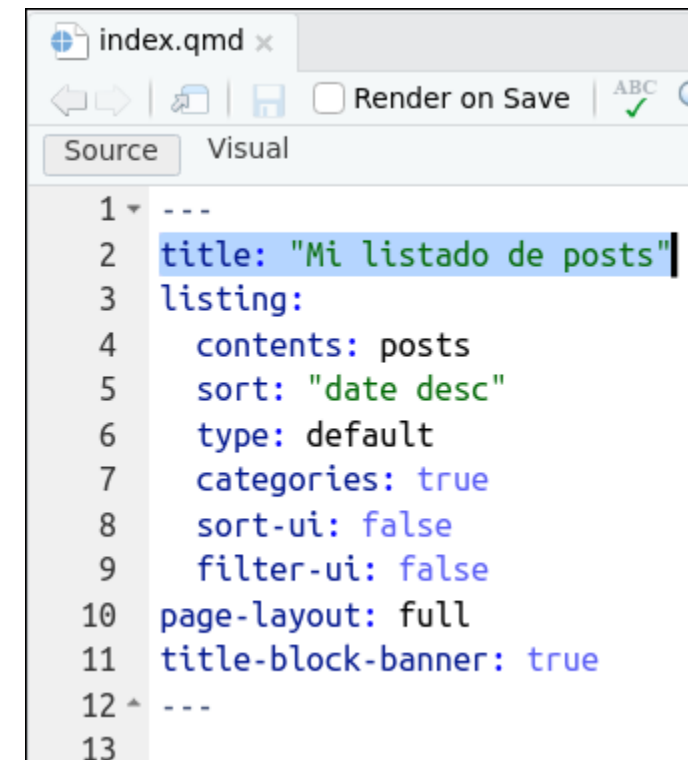
# Archivo `index.qmd`



A screenshot of a code editor showing the original content of `index.qmd`. The editor has a tab labeled 'index.qmd' and a toolbar with icons for undo, redo, save, and a 'Render on Save' checkbox. Below the toolbar are two tabs: 'Source' (selected) and 'Visual'. The source code is as follows:

```
1 ---
2 title: "blog_prueba_00"
3 listing:
4   contents: posts
5   sort: "date desc"
6   type: default
7   categories: true
8   sort-ui: false
9   filter-ui: false
10 page-layout: full
11 title-block-banner: true
12 ---
13
```

Contenido original de `index.qmd`



A screenshot of a code editor showing the modified content of `index.qmd`. The editor has a tab labeled 'index.qmd' and a toolbar with icons for undo, redo, save, a 'Render on Save' checkbox, and a search icon. Below the toolbar are two tabs: 'Source' (selected) and 'Visual'. The source code is as follows:

```
1 ---
2 title: "Mi listado de posts"
3 listing:
4   contents: posts
5   sort: "date desc"
6   type: default
7   categories: true
8   sort-ui: false
9   filter-ui: false
10 page-layout: full
11 title-block-banner: true
12 ---
13
```

Así dejaremos el fichero `index.qmd`

- De momento, solo vamos a hacer cambios en la segunda línea: cambiaremos el título.
- Fíjate que es el título del listado de posts (del blog), no de la página web.
- Fijaros que `index.qmd` es un archivo especial: “sólo tiene yaml”. Es el que genera el listado de posts: es una “listing page”

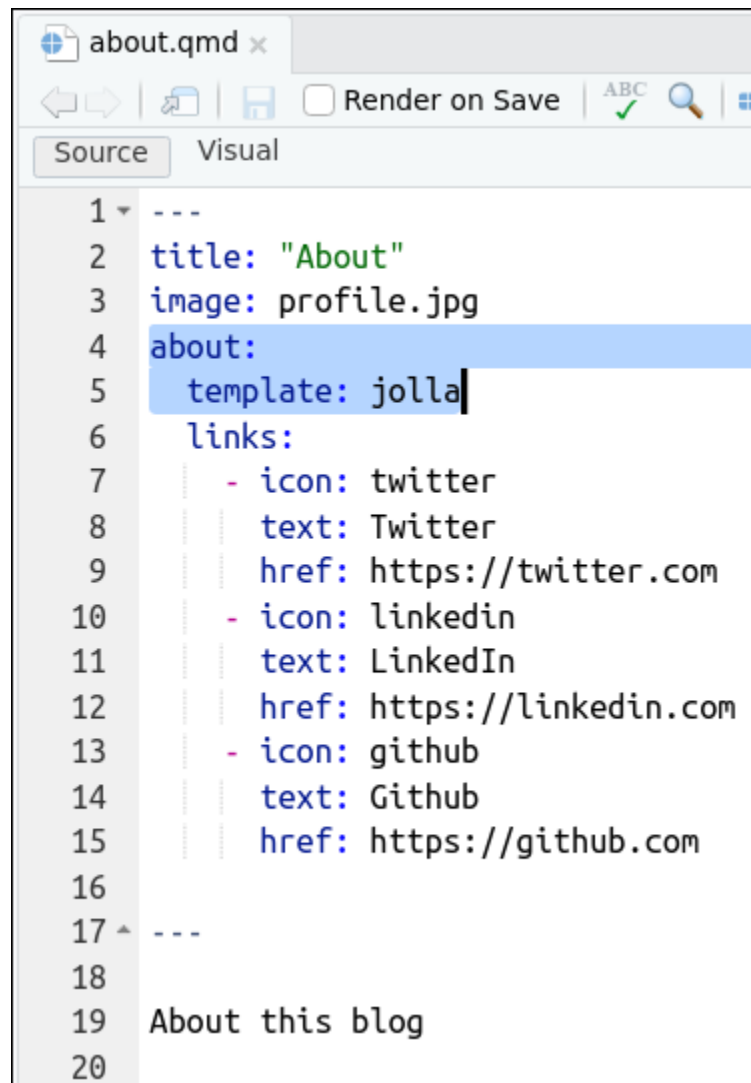


## Tarea 2.5: Modificar el contenido de `index.qmd`

- He cambiado la segunda línea (**title:**) para poner un nombre a la página con el listado de posts.
- También podemos añadir una descripción (con **description:**).

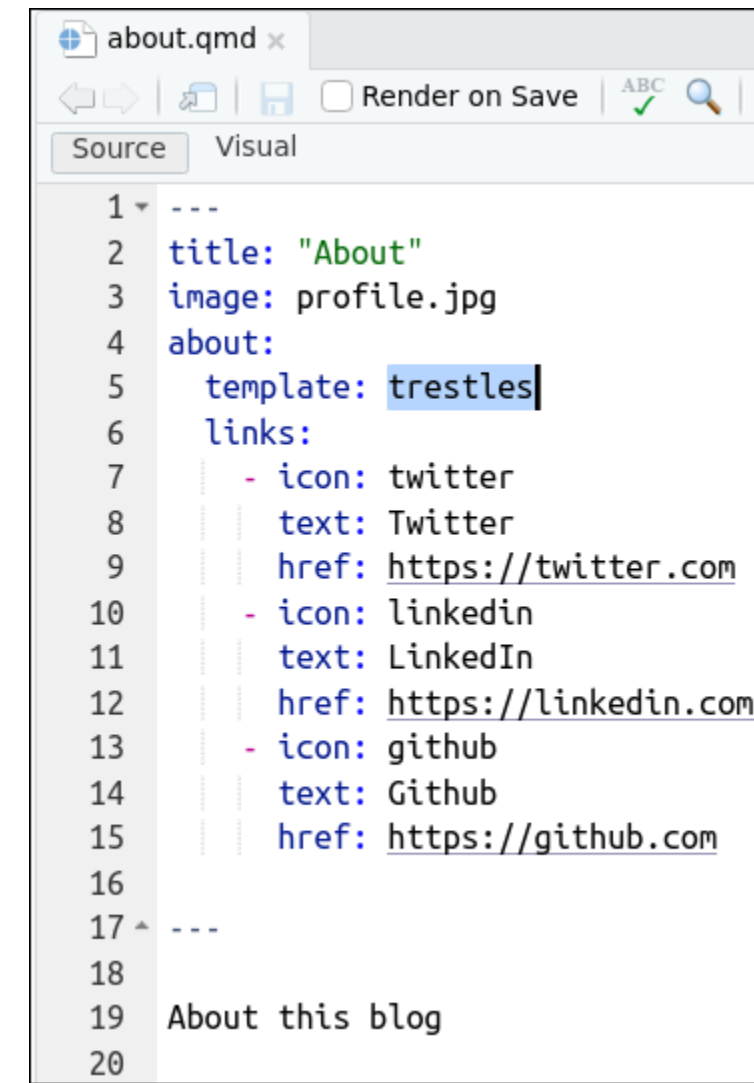
```
1 ---
2 title: "Mi listado de post"
3 description: |
4   Estoy tratando de hacer mi blog
5 listing:
6   contents: posts
7   sort: "date desc"
8   type: default
9   categories: true
10  sort-ui: false
11  filter-ui: false
12 page-layout: full
13 title-block-banner: true
14 ---
```

# Archivo `about.qmd`



```
1 ---
2 title: "About"
3 image: profile.jpg
4 about:
5   template: jolla
6   links:
7     - icon: twitter
8       text: Twitter
9       href: https://twitter.com
10    - icon: linkedin
11      text: LinkedIn
12      href: https://linkedin.com
13    - icon: github
14      text: Github
15      href: https://github.com
16
17 ---
18
19 About this blog
20
```

Contenido original de `about.qmd`



```
1 ---
2 title: "About"
3 image: profile.jpg
4 about:
5   template: trestles
6   links:
7     - icon: twitter
8       text: Twitter
9       href: https://twitter.com
10    - icon: linkedin
11      text: LinkedIn
12      href: https://linkedin.com
13    - icon: github
14      text: Github
15      href: https://github.com
16
17 ---
18
19 About this blog
20
```

Así dejaremos el fichero `about.qmd`

## Tarea 2.6: Modificar el contenido de `about.qmd`

- Antes he dicho que solo modificaríamos la línea 5, pero mejor si lo complementamos con algunas acciones más, como el tamaño de la foto y su forma, por eso he añadido 2 líneas tras `template:trestles`.
- Lógicamente estaría bien que sustituyeses la foto: se trata de cambiar `profile.jpg`, en la línea 3, por la ruta a tu foto.
- Abajo tienes el código:

```
1 ---
2 title: Yo mismo
3 image: profile.jpg
4 about:
5   template: trestles      #- {jolla, solana, ...}
6   image-width: 10em      #- https://cybmeta.com/em-y-rem
7   image-shape: round     #- rectangle, round, rounded
8   links:
9     - icon: twitter
10       text: Twitter
11       href: https://twitter.com
12     - icon: linkedin
13       text: LinkedIn
14       href: https://linkedin.com
15     - icon: github
16       text: Github
17       href: https://github.com
18 ---
19
20 Hola, soy "Yo mismo".
21
22 Estoy asistiendo al curso [**Escribiendo con R y Quarto**] (https://perezp44.github.io/intro.to.quarto/) impartido en en la Universitat de València en 2023
23
24 En un ratito ya piloto Quarto y genero my blog!!! 😊
```



# Archivo `styles.css`

- Puedes cambiar la apariencia estética del blog usando los `themes` predefinidos en Quarto o puedes usar el archivo `styles.css`<sup>1</sup>

## Tarea 2.7: Modificar `styles.css`

- Simplemente sustituye el contenido original del fichero `styles.css` por el contenido del siguiente chunk:

```
1 /* css styles */
2
3
4 /* justificar el texto */
5 p, ul, ol {
6   text-align: justify; }
```

## Resumen (tutorial nº 2)

- En esta tercera parte del taller hemos aprendido a **mejorar/tunear un poco nuestro blog**.
  - Hemos modificado 3 archivos:
    1. `_quarto.yml`: hemos modificado el título del blog y la estructura de la barra de navegación; concretamente hemos añadido una página nueva.
    2. `index.qmd`: construye la Home Page de nuestra web. La opción `listing`: hace que nuestra web es en realidad un blog.
    3. `about.qmd`: hemos utilizado las plantillas de Quarto para configurar nuestra About page.
  - Finalmente hemos modificado (mínimamente) el archivo `styles.css`.
- 
- [Aquí](#) puedes ver como quedaría el blog tras haber modificado `_quarto.yml`, `index.qmd`, `about.qmd` y `styles.css`.

# Práctica (tutorial nº 2)

En esta sección dedicaremos un tiempo a la **práctica libre para que adaptes el blog a tus necesidades y gustos**. Por ejemplo puedes probar:

1. Añadir una **nueva página a la web** ([\\_quarto.yml](#)). Os doy una posibilidad en la siguiente slide
2. Modificar la **apariencia del blog** (realmente del listado de posts) jugando con las opciones que nos proporciona Quarto ([index.qmd](#)). Gracias YAML intelligence!!
3. Modificar la apariencia de la **página About** ([about.qmd](#))

## Práctica 2.1: añadir una página más a nuestra web (`_quarto.yml`)

Evidentemente puedes probar a añadir la página que quieras, pero te ofrezco la siguiente posibilidad:

- Con las líneas 12 y siguientes crear una página más en nuestro blog. La página se llamará “Más cosas”. Además, esta página nueva tiene la particularidad de que cuando recreemos el blog se convertirá en un desplegable con acceso a varias páginas: “Docencia 2”, “Docencia 3” y “Docencia 4”.
- Fíjate que he puesto la nueva página a la izquierda (`left:`)
- Además, puedes probar distintas opciones de `theme` (línea 34) y de `highlighting` (línea 38)
- También puedes cambiar el color del fondo de la barra de navegación (línea 8)

```
1 project:
2   type: website
3
4 website:
5   title: "Mi blog (aún en pruebas)"
6   navbar:
7     logo: "profile.jpg"
8     #background: "#CCCCFF"      #- {"primary", "secondary", "danger", "warning", "light", "dark", or hex color} (el background de la navbar)
9     left:
10      - text: "Docencia"
11        href: docencia.qmd
12      - text: "Más cosas"
13        menu:
14          - text: "Docencia 2"
15            href: docencia.qmd
16          - text: "Docencia 3"
17            href: docencia.qmd
18          - text: "---"
19          - text: "Docencia 4"
20            href: docencia.qmd
21      right:
22        - about.qmd
23        - icon: github
```

```
24 href: https://github.com/<tu-usuario-github>  
25 - icon: twitter  
26 href: https://twitter.com/<tu-usuario-twitter>
```

## Práctica 2.2: Modificar la apariencia del blog (`index.qmd`)

- Se trata de cambiar las opciones predefinidas de tu archivo `index.qmd`
- Con ello podremos cambiar la **apariencia del listado de posts**
- Para ello tienes que ir jugando con las opciones que nos ha proporcionado la plantilla de Quarto en el fichero `index.qmd`.
- la YAML intelligence te ayudará!! (vamos el autocompletado en el YAML)
- No uses las opciones llamadas `custom`

```
1 ---
2 title: "Mi listado de posts"
3 description: |
4   Estoy tratando de hacer mi blog
5 listing:
6   contents: posts
7   sort: "date desc"
8   type: default      #- {default, table, grid, custom}
9   categories: true
10  sort-ui: false
11  filter-ui: false
12 page-layout: full    #- {article, full, custom}
13 title-block-banner: true #- {true, false}
14 ---
```

## Práctica 2.3: Modificar la página About (**about . qmd**)

- Se trata de cambiar las opciones predefinidas de tu archivo **about.qmd**
- Estaría bien que pusieses una foto tuya.
- Poner los enlaces a tus redes sociales y email
- Puedes adaptar el texto y dar algo de formato a la página.

```
1  ---
2  title: Yo mismo
3  image: profile.jpg
4  about:
5    template: trestles      #- {jolla, solana, ...}
6    image-width: 10em      #- https://cybmeta.com/em-y-rem
7    image-shape: round     #- rectangle, round, rounded
8    links:
9      - icon: twitter
10        text: Twitter
11        href: https://twitter.com
12      - icon: linkedin
13        text: LinkedIn
14        href: https://linkedin.com
15      - icon: github
16        text: Github
17        href: https://github.com
18  ---
19
20  Hola, soy "Yo mismo".
21
22  Estoy asistiendo al curso [**Escribiendo con R y Quarto**] (https://perezp44.github.io/intro.to.quarto/) impartido en en la Universitat de València en 2023.
23
24  En un ratito ya piloto Quarto y genero my blog!!! 😊
```

# 3. Convertir el blog en web personal

(3ª parte del tutorial)

## ¿Qué haremos en este tutorial?

- Convertir nuestro blog en una web personal.
- Haremos que la “landing page” o Home de nuestra web no sea una página con el listado de posts, sino que sea una página en la que se reciba a los futuros visitantes de nuestra web
- Para ello solo tendremos que cambiar el nombre de dos ficheros y referenciar uno de ellos en `_quarto.yml`.



### Tarea 3.1: Renombrar el archivo index.qmd

- Cambia el nombre del fichero `index.qmd` a `blog.qmd`
- Ahora este archivo no está referenciado en `_quarto.yml`, por lo que no se mostraría si hacemos un rebuild<sup>1</sup>.

## Tarea 3.2: Incluir `blog.qmd` en la web

- Para que `blog.qmd`, la página que contiene el listado de posts, se muestre en nuestra web hay que referenciarlo/incluirlo en `_quarto.yml`
- Simplemente, tendríamos que modificar las líneas 8 y 9 que ahora están referenciando al fichero `about.qmd`. Por ejemplo podríamos **dejar las líneas 8 y 9 como se ve abajo** (a la derecha):

```
1 #- 2/ Opciones referentes a la estructura de la web -----
2
3 website:
4   title: "Mi blog (aún en pruebas)"
5   navbar:
6     logo: "profile.jpg"
7     right:
8       - text: "Mi blog"
9         href: blog.qmd
10      - icon: github
11        href: https://github.com/<tu-usuario-github>
12      - icon: twitter
13        href: https://twitter.com/<tu-usuario twitter>
14      - icon: envelope
15        url: "mailto:<tu-mail@uv.es>"
16     left:
17       - text: "Docencia"
18         href: docencia.qmd
```

- Para hacerlo, es mejor copiar el código de abajo y pegarlo en `_quarto.yml`

```
1 project:
2   type: website
3
```

```

4 website:
5   title: "Mi blog (aún en pruebas)"
6   navbar:
7     logo: "profile.jpg"
8     #background: "#CCCCFF"    #- {"primary", "secondary", "danger", "warning", "light", "dark", or hex color} (el backgroung de la navbar)
9     left:
10      - text: "Docencia"
11        href: docencia.qmd
12      - text: "Más cosas"
13        menu:
14          - text: "Docencia 2"
15            href: docencia.qmd
16          - text: "Docencia 3"
17            href: docencia.qmd
18          - text: "---"
19          - text: "Docencia 4"
20            href: docencia.qmd
21      right:
22        - text: "Mi blog"
23          href: blog.qmd
24        - icon: github
25          href: https://github.com/<tu-usuario-github>
26        - icon: twitter

```

- Aún no podemos refrescar la web porque ahora no tenemos archivo `index.qmd`.

### Tarea 3.3: Renombrar el archivo about.qmd

- Cambia el nombre del fichero `about.qmd` a `index.qmd`, con lo que pasará ser nuestra “landing page”.
- Como el antiguo fichero `about` contenía nuestra foto parece que es un buen candidato a ser nuestra “Home”

### Tarea 3.4: Hacemos un rebuild de la web

- Ahora ya tenemos fichero `index.qmd`, por lo que ya podemos hacer un rebuild de nuestra web y veremos la nueva estructura de nuestra web, con una foto nuestra en la “landing page”.
- Para ello: **Build > Render Website** en RStudio.

# 4. Workflow: ¿cómo crear un post?

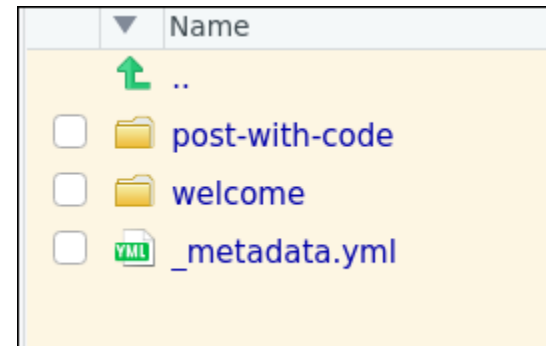
(4ª parte del tutorial)

¿Qué haremos en este tutorial?

- Ver como podemos iniciar un nuevo post
- (podemos marcar post todavía no acabados con `draft: true` en el yaml del post)

# ¿Donde están los posts del blog?

- Los ficheros `.qmd` que generan los posts están en la carpeta **posts** (de nuestro Qproject que genera el blog).
- Cada post está en una carpeta diferente<sup>1</sup>:



- Veamos (en la siguiente slide) el contenido de `./posts/welcome/`

# Contenido de cada subcarpeta de `./posts/`

- Por ejemplo de `./posts/welcome/`

	Name	Size
	..	
<input type="checkbox"/>	index.qmd	297 B
<input type="checkbox"/>	thumbnail.jpg	39 KB

Contenido de la carpeta `welcome` que genera el post titulado “Welcome to my blog”

```
1 ---
2 title: "Welcome To My Blog"
3 author: "Tristan O'Malley"
4 date: "2022-10-28"
5 categories: [news]
6 ---
7
8 This is the first post in a Quarto blog. Welcome!
9
10 
11
12 Since this post doesn't specify an explicit `image`, the first image
13 in the post will be used in the listing page of posts.
14
15
```

Contenido del archivo `index.qmd`




## Tarea 4.1: Creando un nuevo post

- Crea una nueva subcarpeta en `./posts/`
- Ponle nombre, por ejemplo: “my-primer-post”
- Copia en la carpeta que hemos creado el contenido de la carpeta de uno de los 2 posts. Por ejemplo copia los 2 archivos que hay en `./posts/welcome/` y pégalos en `./posts/my-primer-post/` ; es decir, pegamos el archivo `index.qmd` y `thumbnail.jpg` en nuestra nueva carpeta.
- Vuelve a regenerar el blog: `Build > Render Website` y verás que ahora nuestro blog ya tiene tres posts.

## Tarea 4.2: Modificando el tercer post

- Modifica el contenido de `/.posts/my-primer-post/index.qmd`. Por ejemplo, puedes dejarlo así:

```
1 ---
2 title: "Mi primer post con Quarto"
3 author: "Pedro J. Pérez"
4 date: "2022-10-25"
5 categories: [R, quarto]
6 image: "thumbnail.jpg"
7 ---
8
9 Este es mi primer post con Quarto!!!!
10
11 ```{r}
12 sqrt(2 + 2)
13 ```
14
15 Voy a poner una imagen:
16
17 
18
19
20 Acabé con mi primer post. FIN
21
```

- Sustituye `thumbnail.jpg` por otro archivo
- La opción de yaml (línea 6): `image: "thumbnail.jpg"` nos permite elegir la imagen que acompañara al post en la listing page.
- Vuelve a regenerar el blog: `Build > Render Website`

## Resumen (tutorial nº 4)

- En esta **cuarta parte del taller** hemos visto como podemos generar/escribir nuevos posts para nuestro blog: solo hay replicar una de las carpetas de anteriores posts y cambiarle el nombre.
- Hemos visto como cambiar la imagen que acompaña a los posts en la “listing page”.
- Recuerda que podemos marcar un post con `draft: true` en el yaml del post.
- En el **siguiente tutorial** veremos en profundidad las posibilidades que nos ofrece Quarto para escribir nuestros posts.

# 5. Escribiendo posts: practicando con Quarto

(5ª parte del tutorial)

¿Qué haremos en este tutorial?

- Aprender a escribir nuestro posts con Quarto: con ficheros `.qmd`
- Repasaremos las 3 partes de los documentos `.qmd`: yaml, texto, chunks
- Además veremos: **layout**
- Aprender a utilizar algunos truquillos de Quarto

## Advertencia

- ~~Voy a suponer que los asistentes al taller, y posibles lectores de estos tutoriales, conocen Rmarkdown~~
- El paso de `.Rmd` a `.qmd` es sencillo.
- En cualquier caso, [aquí](#) tienes unas FAQ's para usuarios de `.Rmd` que se preguntan si pasarse a Quarto, y [aquí](#) unas slides sobre las diferencias entre `.qmd` y `.Rmd`.
- [Aquí](#) y [aquí](#) slides sobre cómo convertir ficheros `.Rmd` a `.qmd`
- Aquellos que no sepan qué es Rmarkdown pueden empezar por [aquí](#) o [aquí](#).

## En realidad la advertencia no aplica a los asistentes al “curso de intro a Quarto”

- Lógicamente, a los asistentes al curso no os hace falta conocer Rmarkdown, porque ya estáis aprendiendo Quarto, la segunda generación de `.Rmd`
- Ya vimos con las [Slides 03: Aprendiendo a escribir en QMD](#) cómo escribir con `.qmd`

# Ficheros `.qmd`

- Los ficheros `.qmd` tienen 3 partes: YAML, texto y chunks de código.

```
1 ---
2 title: "Mi primer post con Quarto"
3 author: "Pedro J. Pérez"
4 date: "2022-10-25"
5 categories: [R, Quarto]
6 ---
7
8 Este es mi **primer post con Quarto!!!
9
10 ```{r}
11 2 + 2
12 ```
13
14 Acabé mi primer post. FIN
15
```

- Veámoslas una a una

# 1. YAML

# YAML: ideas importantes

- El encabezamiento o **YAML** sirve para fijar **opciones y metadatos** de nuestro documento.
- El **YAML** será procesado varias veces durante el procesamiento del documento: es leído por Quarto, knitr y Pandoc e **influirá en el output final**.
- El hecho de estar trabajando dentro de un **Qproject** nos da **mucha versatilidad** a la hora de especificar el YAML de nuestros documentos **.qmd**. Documentación oficial [aquí](#)



# El `yaml` de un post se puede especificar en 3 niveles

1. **Nivel proyecto:** todo Qproject tiene archivo `_quarto.yaml`. Todo documento que se procese dentro del proyecto, heredará los metadatos definidos en `_quarto.yaml`.
2. **Nivel carpeta:** si en una carpeta existe un documento `_metadata.yaml`, los documentos de esa carpeta heredan sus metadatos. La carpeta `./posts/` de un blog suele tener un archivo `_metadata.yaml`.
3. **Nivel documento:** en el yaml del propio documento `.qmd`
  - Si hay conflictos **prevalecen las opciones del nivel documento**, luego nivel carpeta y finalmente nivel proyecto.
  - **Documentación oficial** de Quarto con las **principales opciones** que se pueden fijar en el YAML para documentos html: [aquí](#) y [aquí](#)

# YAML: tal como lo tenemos ahora

1. nivel proyecto (\_quarto.yml)

2. nivel carpeta (\_metadata.yml)

3. nivel documento (index.qmd)

Abajo el contenido del fichero `_quarto.yml` tal y como lo tenemos ahora en nuestro blog. Solo he añadido 3 comentarios para diferenciar entre secciones:

```
1 # 1/ Opciones referentes al proyecto -----
2 ▾ project:
3   type: website
4
5 #- 2/ Opciones referentes a la estructura de la web -----
6 ▾ website:
7   title: "Mi blog (aún en pruebas)"
8 ▾ navbar:
9   logo: "profile.jpg"
10 ▾ right:
11 ▾   - text: "About me"
12   href: about.qmd
13 ▾   - icon: github
14   href: https://github.com/<tu-usuario-github>
15 ▾   - icon: twitter
16   href: https://twitter.com/<tu-usuario-twitter>
17 ▾   - icon: envelope
18   url: "mailto:<tu-mail@uv.es>"
19 ▾ left:
20 ▾   - text: "Docencia"
21   href: docencia.qmd
22 |
23 #- 3/ Opciones referentes al formato de salida -----
24 ▾ format:
```

# YAML: NIVEL PROYECTO (opciones en `_quarto.yml`)

En `_quarto.yml` se suelen poner opciones referentes a 3 aspectos:

1. Sobre el propio Qproject

2. Estructura de la página web

3. Formato de salida de los documentos


```
1 # 1/ Opciones referentes al proyecto -----
2 project:
3   type: website
4   output-dir: docs    #- podemos cambiar la carpeta donde se redirige el output (.html)
5   # execute-dir: project    #- cambiar el render directory: https://quarto.org/docs/projects/code-execution.html#working-dir
```





- línea 4: podemos elegir la carpeta de destino de nuestro blog
- línea 5: podemos cambiar el render directory de los `.qmd`

Documentación oficial [aquí](#).

# Un ejemplo “completito” de `_quarto.yml`

Si quieres ver un documento `_quarto.yml` completito, ve [aquí](#). Pertenece a la [web de Quarto](#).





Guide

## Guide

Comprehensive guide to using Quarto. If you are just starting out, you may want to explore the [tutorials](#) to learn the basics.

Authoring	Computations	Tools
Create content with markdown	Execute code and display its output	Use your favorite tools with Quarto
<a href="#">Markdown Basics</a>	<a href="#">Using Python</a>	<a href="#">JupyterLab</a>
<a href="#">Figures</a>	<a href="#">Using R</a>	<a href="#">RStudio IDE</a>
<a href="#">Tables</a>	<a href="#">Using Julia</a>	<a href="#">VS Code</a>
<a href="#">Diagrams</a>	<a href="#">Using Observable</a>	<a href="#">Neovim</a>

```
1 project:
2   type: website
3   output-dir: _site
4   resources:
5     - "/docs/download/_download.json"
6     - "/docs/download/_prerelease.json"
7     - "/_redirects"
8
9 website:
10  title: "Quarto"
11  image: "quarto.png"
12  favicon: "favicon.png"
13  google-analytics: "G-FV9Z7SDZ0M"
14  open-graph: true
15  twitter-card: true
16  site-url: https://quarto.org
17  repo-url: https://github.com/quarto-dev/quarto-web
18  repo-actions: [edit, issue]
19  page-navigation: true
20  search:
21    algolia:
22      index-name: prod_QUARTO
23      application-id: ZPJB5I1QN7
24      search-only-api-key: 41be6c1e0a7fea4a51b107810facf577
25      analytics-events: true
26      show-logo: true
```

## Tarea 5.1: Modificar `_quarto.yml`

- Tenemos que dejar el archivo `_quarto.yml` así:

```
1 # 1/ Opciones referentes al proyecto -----
2 project:
3   type: website
4   output-dir: _site
5
6 #- 2/ Opciones referentes a la estructura de la web -----
7 website:
8   title: "Mi blog (aún en pruebas)"
9   navbar:
10     logo: "profile.jpg"
11     #background: "#CCCCFF"      #- {"primary", "secondary", "danger", "warning", "light", "dark", or hex color} (el background de la navbar)
12     right:
13       - text: "Mi blog"
14         href: blog.qmd
15       - icon: github
16         href: https://github.com/<tu-usuario-github>
17       - icon: twitter
18         href: https://twitter.com/<tu-usuario-twitter>
19       - icon: envelope
20         url: "mailto:<tu-mail@uv.es>"
21     left:
22       - icon: house-door
23         href: index.html
24       - text: "Docencia"
25         href: docencia.qmd
26       #- text: "Docencia"
27       #- text: "Contacto"
```

# YAML nivel carpeta

- Abajo el contenido del fichero `./posts/_metadata.yml` de nuestro blog

```
1 # options specified here will apply to all posts in this folder
2
3 # freeze computational output
4 # (see https://quarto.org/docs/projects/code-execution.html#freeze)
5 freeze: true
6
7 # Enable banner style title blocks
8 title-block-banner: true
9
```

# YAML nivel carpeta

## Tarea 5.2: Modificar `./posts/_metadata.yml`

- Tenemos que dejar el archivo `./posts/_metadata.yml` así:

```
1 # IMPORTANTE: options specified here will apply to all posts in this folder
2
3 # freeze: controla la ejecución de los .qmd durante un global project render
4 # https://quarto.org/docs/projects/code-execution.html#freeze)
5 freeze: auto    #- {false, true, auto}
6
7 #- Quarto incluye un title-block al principio de los artículos con elementos como: title, subtitle, authors, date, doi, and abstract.
8
9 # title-block-banner: pone un banner en el title-block: pondrá el title, subtitle, description, y categories dentro del banner
10 # https://quarto.org/docs/authoring/title-blocks.html#title-banners
11 title-block-banner: true  #- {true, false, "#FFDDFF", "image.jpg"}
12
13 # title-block-style: modifica el estilo del title-block
14 # https://quarto.org/docs/authoring/title-blocks.html
15 title-block-style: default  #- {default, plain, none}
16
17 # tb se puede especificar el color del texto dentro del banner
18 title-block-banner-color: red
```

**IMPORTANTE:** las opciones que especifiques en `_metadata.yml` afectarán a todos los `.qmd` de la carpeta.

- `freeze` (línea 5): `{false, true, auto}` controla la ejecución de los `.qmd` durante el procesado completo del Qproject. Documentación [aquí](#).
- `title-block-banner` (línea 11): `{true, false, "#FFDDFF", "image.jpg"}` controla la presencia y apariencia de los “banners” para dar más importancia a los títulos de los posts. Puedes controlar el color, incluso poner una imagen. Documentación [aquí](#).
- `title-block-style` (línea 15): `{default, plain, none}` modifica el estilo del title-block
- `title-block-banner-color` (línea 18): `{black, "#FFDDFF"}` controla el color del título del post



# YAML: nivel documento

## Tarea 5.3: Modificar encabezamiento de `./posts/my-primer-post/index.qmd`

- Tenemos que dejar el archivo `./posts/my-primer-post/index.qmd` así:

```
1 ---
2 title: "Mi primer post con Quarto"
3 description: |
4   Estoy aprendiendo Quarto
5 author:
6   - name: Pedro J. Pérez
7     url: https://www.wikidata.org
8     affiliation: Universitat de València
9     affiliation-url: https://www.uv.es
10    orcid: 0000-xxxx
11 date: "2022-10-25"
12 categories: [R, quarto]
13 #title-block-banner: true #- {true, false }
14 title-block-banner: thumbnail.jpg
15 title-block-banner-color: green
16 ---
17
18 Este es mi primer post con Quarto!!!!
19
20 ```{r}
21 sqrt(2 + 2)
22 ```
23
24 Voy a poner una imagen:
25
26 
27
```

- Sustituye `thumbnail.jpg` por otro archivo
- Vuelve a regenerar el blog: **Build > Render Website**  
<https://go.uv.es/pjperez/intro.to.quarto>

## 2. TEXTO (o narrativas)

# Texto (o narrativas)

- Se escribe (al igual que `.Rmd`) en **markdown**. [Aquí](#) la documentación oficial de Quarto.
- Sintaxis básica de `markdown`

SI ESCRIBES:

```
# Esto es un título de primer nivel
```

```
## Esto es un título de segundo nivel
```

```
### A. Formateo de texto
```

Esto es texto normal

```
**Esto se mostrará en negrita**
```

```
*Esto se mostrará en cursiva*
```

```
~~Esto se mostrará tachado~~
```

```
> Esto se mostrará como una cita
```

```
### B. Listas
```

Listas con números:

1. Primero
2. Segundo
3. Tercero

Listas con bullets:

- Primero
- segundo
- Tercero

Mas listas:

1. Primero
  - 1.1
  - 1.2
2. Segundo
  - a
  - b
3. Tercero

SE MOSTRARÁ:

# Esto es un título de primer nivel

## Esto es un título de segundo nivel

### A. Formateo de texto

Esto es texto normal

**Esto se mostrará en negrita**

*Esto se mostrará en cursiva*

~~Esto se mostrará tachado~~

Esto se mostrará como una cita

### B. Listas

Listas con números:

1. Primero
2. Segundo
3. Tercero

Listas con bullets:

- Primero
- segundo
- Tercero

Mas listas:

1. Primero
  - o 1.1
  - o 1.2
2. Segundo
  - o a
  - o b
3. Tercero

# 3. CHUNKS

# CHUNKS

Comportamiento similar a los documentos `.Rmd`. La documentación oficial está [aquí](#)

## Principales diferencias con `.Rmd`

- En ficheros `.qmd`, las opciones de los chunks se pueden especificar globalmente en el YAML y a nivel individual en cada uno de los chunks.
- En los **chunks individuales** ahora se utiliza la **sintaxis YAML** (**key: value**) en líneas dentro del chunk que empiezan con `#|`. Por ejemplo:

```
1 ---
2 title: "Mi segundo post"
3 execute:
4   echo: false
5   eval: true
6 ---
7
8 Este es mi **segundo** post con Qu
9
10 ```{r}
11 #| eval: false
12 2 + 2
13 ```
14
15 ```{r}
16 summary(mtcars)
17 ```
```

# CHUNKS

- Las principales opciones son: **echo**, **eval**, **warning**, **error**, **output** e **include**. [Aquí](#) la documentación oficial.
- **echo**: además de los típicos **true** y **false**, ahora **incorpora un nuevo valor **fenced**** que facilita mostrar las marcas de los chunks en el documento final. Documentación [aquí](#).
- Además, si usamos **knitr** para ejecutar los chunks, entonces podemos usar todas las **opciones nativas de knitr**, como: **collapse**, **fig.width**, **comment**, etc ... Más información [aquí](#). Un ejemplo:

```
1 ---
2 title: "Mi segundo post"
3 author: "Pedro J. Pérez"
4 execute:
5   echo: fenced
6 knitr:
7   opts_chunk:
8     collapse: false
9     comment: "#>"
10 ---
11
12 Este es mi **segundo** post
13
```

```
1 ---
2 title: "Mi segundo post"
3 author: "Pedro J. Pérez"
4 execute:
5   echo: fenced
6 knitr:
7   opts_chunk:
8     collapse: false
9     comment: "#>|"
10 ---
11
12 Este es mi **segundo** post
13
```

- Hay más opciones para los chunks. Por ejemplo:
  - hacer **folding code** con `#| code-fold: true`
  - si el código es muy largo, puedes usar `#| code-overflow: wrap` o `scroll`
  - puedes hacer que se muestren los **números de línea** con `#| code-line-numbers: true`

La documentación oficial la tienes [aquí](https://go.uv.es/pjpperez/intro.to.quarto).



# 4. Elementos básicos para escribir

# Elementos básicos para escribir

Para escribir en Markdown, además de texto, tenemos los siguientes elementos básicos:

- Links (documentación [aquí](#))
- Listas (documentación [aquí](#) y [aquí](#))
- Tablas (documentación [aquí](#))
- Código (documentación [aquí](#))
- Ecuaciones (documentación [aquí](#)). Editor de ecuaciones [aquí](#) o [aquí](#).
- Imágenes (documentación [aquí](#)) y sus [opciones](#)

El [editor visual](#) facilita la introducción de estos elementos en tus documentos. En [esta charla](#) de Mine Çetinkaya-Rundel se aprecia fácilmente la **utilidad del editor visual**.

# 5. Más elementos para “escribir”

# Más elementos para “escribir”

Además de los elementos básicos de escritura que nos ofrece la sintaxis básica de `markdown`, Quarto nos ofrece más posibilidades. Por ejemplo:

- **Layout** (documentación [aquí](#) y [aquí](#))
- **Imágenes** (documentación [aquí](#)) y sus [opciones](#)
- **Extensiones** (documentación [aquí](#)) . Listado de extensiones oficiales [aquí](#)
- **Shortcodes** (documentación [aquí](#))
- **Callout blocks** (documentación [aquí](#))
- **Divs & spans** (documentación [aquí](#))

Veamos algunos de estos elementos con un poco de detalle. Después lo recordaremos con una Práctica.

# **Veamos algunos de estos elementos con un poco de detalle.**

(Después lo recordaremos con una Práctica)

# Layout (“general”)

- Además de yaml, texto y chunk, para escribir con Quarto **conviene conocer la estructura o layout de las páginas web** que generamos con Quarto. Documentación [aquí](#) y [aquí](#).
- Nuestra páginas web tienen 2 zonas si la miramos en horizontal: la **sidebar** y la zona reservada para **el artículo**. Puedes verlo [aquí](#)
  - A su vez, la zona del artículo se divide en 2: **body** y **margin**.
- A partir de la versión 1.3 de Quarto, se puede controlar estos componentes del layout. Puedes verlo [aquí](#)
- Además, el layout puede venir afectado por la opción **page-layout: full** que tiene 3 opciones: article, full y custom

# Layout's (“para secciones específicas”)

⚠ Importante!!

- Usando fenced-divs se puede hacer que **alguna sección de nuestra página ocupe un espacio mayor al habitual**; es decir, mayor al espacio habitualmente reservado para el “body”.
- De esta forma podemos hacer que, por ejemplo, una imagen<sup>1</sup> ocupe más espacio del habitual, más espacio del reservado al body, o incluso desplazarla al “margin”.
- Veamos las **posibilidades que tenemos de layout** con [este post](https://go.uv.es/pjperez/intro.to.quarto) del blog del curso

# Imágenes

- Conocemos ya la sintaxis básica para insertar imágenes ! `[](<ruta-a-imagen>)`; ahora veremos posibilidades más avanzadas
- Veamos como **insertar imágenes** con [este post](https://go.uv.es/pjpperez/intro.to.quarto) del blog del curso



# Extensiones

- Una de las novedades de Quarto es la posibilidad de usar **extensiones**. Las extensiones se escriben en lenguaje [Lua](#).
- Para ver como **instalar y usar extensiones** vamos a ir, otra vez, a otro de los post del blog del curso: concretamente [aquí](#)

# Shortcodes

- Un shortcode es un código (una etiqueta, un elemento) que podemos **insertar** para añadir algún tipo de **contenido sin tener que escribirlo directamente**.
- Son muy **útiles** para insertar **contenido que se repite**
- Podemos verlo en [este post](#) del blog del curso o en la [documentación oficial](#)

# Faltan 2 elementos

- Callout blocks: documentación [aquí](#)
- Divs y spans: documentación [aquí](#)

## Resumen (tutorial nº 5)

- La **quinta parte del taller** se ocupa de la **escritura de posts** para el blog.
- Los posts **se han de escribir en formato `.qmd`**; por lo tanto, se repasan los fundamentos “teóricos” necesarios para escribir este tipo de documentos.
- Nos apoyamos en nuestro ~~conocimiento previo de la sintaxis Rmarkdown~~.
- Se repasan las posibilidades de configurar los documentos por medio de **los 3 niveles de YAML** que pueden afectar a un documento.
- **Los 3 YAML's** son: Nivel proyecto (`_quarto.yml`), nivel carpeta (`_metadata.yml`) y nivel documento (en el propio encabezamiento del archivo)
- El **texto** en ficheros `.qmd` es muy similar al de los documentos `.Rmd`. Aparecen algunos elementos nuevos como los Call-outs.
- El **código** sí presenta novedades: nueva sintaxis con `# |` y posibilidad de determinar opciones de los chunks en el YAML.
- Para poder escribir en `.qmd` **hay que practicar**. Lo haremos con una tarea donde veremos, entre otras cosas:
  - Algunas de las opciones de **layout**.
  - Ver como instalar y usar **extensiones**.
  - Opciones avanzadas para **insertar imágenes**.

## Práctica 5.1: Crear un post para tu blog con la siguiente plantilla.

Vamos a crear un nuevo post para practicar algunos de los elementos de escritura que hemos visto. Para ello:

- Crea una nueva subcarpeta en la carpeta `./posts/`. Por ejemplo llámala: `./posts/my-segundo-post` y copia allí los documentos que haya en la carpeta de otro blog. Es decir, en la nueva subcarpeta que has creado, debe haber un fichero `index.qmd` y una imagen `thumbnail.jpg`.
- Sustituye el contenido del archivo `index.qmd` por el contenido del siguiente chunk y haz un render de `./posts/my-segundo-post/index.qmd` para ver como quedaría una vez procesado.

### Contenido para sustituir en `./posts/my-segundo-post/index.qmd`

```
1 ---
2 title: "Mi Segundo post con Quarto"
3 author: "Nosotros"
4 date: 2022-10-25
5 categories: [R, quarto, ejemplos]
6 image: "thumbnail.jpg"
7 #subtitle: |
8 #   Practicando con Quarto
9 description: |
10   Estamos viendo algunos elementos para escribir con Quarto.
11 ---
12
13 Este ya es mi segundo post con Quarto. En él voy a hacer lo siguiente:
14
15
16 ## Insertar 2 imágenes side-by-side
17
18
19 ## Inserta un tweet
20
21
22 ## Inserta un `tab-set`
23
24
25 ## Inserta un call-out
```

Las soluciones a la Práctica están [aquí](#)

# 6. Tuneado del blog

(6ª parte del tutorial)

¿Qué haremos en este tutorial?

- Ver distintas opciones para adaptar el blog a nuestras necesidades
- En concreto veremos 3 formas de hacerlo: a través de opciones de Quarto en el yaml, con CSS y con SASS

# Intro

- Quarto utiliza el framework [Bootstrap 5](#) al generar sus documentos `.html`.
- Además, incorpora 25 themes del [Bootswatch project](#) que le dan a nuestro blog/web una apariencia profesional y cuidada.
- Creo que es mejor **comenzar con un blog sencillo** usando las plantillas, o **themes**, que nos ofrece Quarto pero, si queremos modificar la apariencia de nuestro blog, podemos hacerlo de **3 formas**:
  1. Utilizando las opciones disponibles en Quarto para el `yaml`
  2. Utilizando CSS
  3. Utilizando SASS



# 1. Utilizando opciones del YAML

- En el [tutorial nº 6](#) (del taller que impartí en Córdoba) repasamos las distintas opciones que tenemos disponibles para poder cambiar a través de opciones en los YAML's (recuerda los 3 niveles).
- Aquí solo pondré la **documentación relevante** junta y **destacaré algunas de las opciones** disponibles.

# Documentación oficial

(sobre opciones disponibles en los YAML's)

- Referentes al **proyecto**: [aquí](#)
- Referentes a la **estructura de la web**: [aquí](#) y [aquí](#)
- Referentes a la **estética**: [aquí](#) y [aquí](#)
- Referentes a las **Listing Pages**: [aquí](#)
- Referentes a las **About Pages**: [aquí](#)

# Algunas opciones de YAML

# Algunas opciones de YAML: lang

- Si te has fijado, los metadatos de los documentos aparecen en inglés.
- Por ejemplo, pone “Author” en lugar de “Autor” o “Autora”.

## Hagamos algunos cambios

- Si quisiéramos cambiar específicamente la opción de autor, tendríamos que poner en el YAML:

```
1 language:  
2   title-block-author-single: "Autora"
```

- Podemos cambiar todas las opciones especificando en el YAML `lang: es`.
- La documentación oficial está [aquí](#) y [aquí](#) todos los elementos que se pueden modificar. Y [aquí](#) los ficheros específicos para diferentes idiomas. [Aquí](#) el documento para el castellano.

- Si quisieras usar tu propio documento tendrías que poner en el YAML: `language: custom.yml` (lógicamente el fichero `custom.yml` tendría que existir y estar en la ruta correcta).
- ¿Probamos a hacerlo?

# Otras opciones para tunear desde el YAML

- **TOC:** es importante tenerlo a nuestro gusto. [Aquí](#) tienes las opciones que puedes ajustar con opciones en el yaml
- **Chunks:** [aquí](#) tienes las opciones que puedes ajustar desde el YAML
- **Resizing de los thumbnails.** Un [gist](#) para hacerlo.
- **Algunas opciones de tuneado** que puedes implementar desde el YAML. Prueba a poner estas opciones en el archivo `_quarto.yml`. Nuestro blog **empeorará bastante**. Más opciones [aquí](#).

```
1 fontcolor: green      #- color del texto
2 linkcolor: purple     #- color de los enlaces
3 monobackgroundcolor: red #- color de los cuadros de resultados de evaluar código
4 fontsize: 0.6em       #- tamaño del texto (más pequeño de lo normal: 1)
5 linestretch: 2.3      #- tamaño entre las líneas (1.6 es lo "normal")
```

## 2. Utilizando CSS

- La apariencia visual del blog puede cambiarse utilizando CSS.
- Veamos un ejemplo con el siguiente [post](#)

# 3. Utilizando SASS

- La documentación oficial [aquí](#)
- [Aquí](#) la documentación oficial de Quarto sobre los Bootswatch Sass Theme Files.
- Bootstrap define unas 1.400 variables con las que controlar fuentes, colores, etc .... Puedes verlas [aquí](#)
- [Aquí](#) están los ficheros `.scss` de los 25 built-in Bootswatch themes.
- **Otras referencias**, por ejemplo: [Customizing Quarto Websites: Make your website stand out using SASS](#) o [este video](#)
- Una pequeña práctica en el [tutorial nº6](#) del taller “Mi primer blog con Quarto”.



# Publicando el blog

- La [documentación oficial](#) de Quarto.
- Si lo publicamos en [Quarto Pub](#), ejecutar en la Terminal: `quarto publish quarto-pub`
- En [Github Pages](#)
- En la Universitat de València.
  - Un post que lo explica [aquí](#)

# Fin!!

- Muchas gracias por la atención 😊
- Espero que el taller haya salido OK ✅
- Si alguien motivado por el taller acaba haciéndose un blog, **please que me avise** (pedro.j.perez@uv.es) 📧
- Big thanks to all the Posit/Quarto team !!!! 🙌🙌🙌
- Bye 🙌🙌 , nos vemos en las próximas Jornadas en ...

