

## Ficha Prática nº 7 - *Galeria de Imagens Flexível*

### Exercício 1: Menu com *flexbox* / CSS Grid

- a) Crie uma pasta designada por **ficha7** e copie os ficheiros do `ficha_7.zip` para essa pasta.
- b) Inicie o *Visual Studio Code* e abra a pasta anterior.
- c) Analise o código do **ficha7.html** bem como os seletores pré-definidos no ficheiro **ficha7\_Menu.css**, identificando a que elementos HTML estes seletores CSS se aplicam.
- d) Como pode verificar a barra de navegação (nav) assim como o rodapé (*footer*) já se encontram formatados.

Note que estes elementos foram implementados com base em *flexbox* o que os capacita para se ajustarem a diferentes dimensões do viewport. Confirme recorrendo às *developer tools*.

- e) Efetue as devidas alterações por forma a que, quando se passa o rato por cima, a cor do texto passe a ser branca, a cor de fundo #8f6e4b (Figura 1). Adicione ainda a propriedade *cursor* com o valor *pointer*.



Figura 1 – menu de navegação (hover)

- f) Visualize a página no browser. Redimensione o *viewport* e verifique que o último elemento se mantém sempre à direita.
- g) Altere o seletor **.menu** que especifica `display:flex` para **.menu:not(.grid) {... }**

**Nota:** O seletor de pseudo class `:not(selector)` permite selecionar todos os elementos que não são afetados por *selector*.

- h) Visualize a página novamente, e verifique que não houve qualquer alteração.
- i) Especifique um seletor que recorra a classe de nome **grid**, a qual deverá ser aplicada apenas ao menu. Adicione a seguinte regra **display: grid**.
- j) Altere o elemento ul cuja classe é menu, de forma a ser aplicada também a classe **grid**.
- k) No seletor definido na alínea i), adicione uma regra de forma a especificar uma **grid**, **constituída por 4 colunas e uma linha** para que o menu fique com o aspeto da Figura1. As 3 primeiras colunas devem ter o tamanho **auto** e a ultima coluna deverá ocupar o restante espaço disponível no *viewport*, portanto, deverá especificar `1fr`.

**Nota:** O valor `auto` define a largura da coluna tendo como base a largura do respetivo conteúdo.

## Exercício 2: Galeria de Imagens

- a) Analise o HTML responsável por apresentar as imagens na página.
- b) Abra o ficheiro **ficha7\_Gallery.css**, e analise o código disponível. Como é possível verificar o elemento *container* ocupa 60% da largura disponível.
- c) Implemente a **grid** que vai constituir a base para a construção da galeria de imagens. Deve construir uma **grid 8x8** (Figura 2) em que as colunas vão aproveitar todo o espaço disponível e as linhas correspondem à dimensão de 5% da largura do *viewport* (*viewport width*). O espaçamento entre células deve ser de 0.2% da largura do *viewport*.

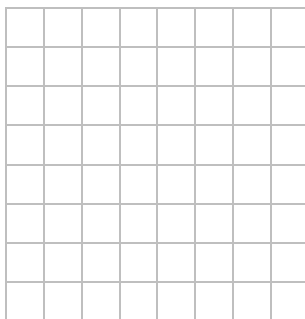


Figura 2 – Grid para posicionamento das imagens

- d) As imagens devem ser posicionadas garantindo que ocupam 100% da respetiva célula da CSS Grid. De forma a garantir que a imagem, apesar de redimensionada, preserva o seu *aspect ratio* original recorra à propriedade **object-fit:cover**.
- e) Posicione as imagens sequencialmente da forma indicada na figura 3.

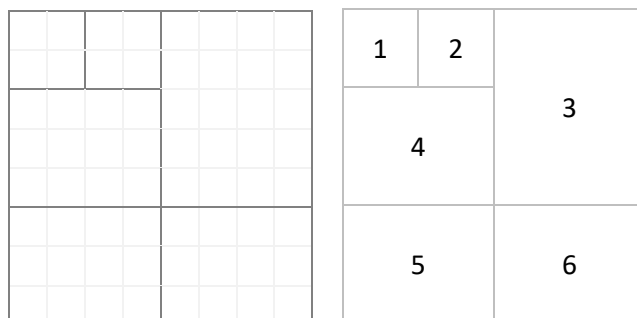


Figura 3 – Posicionamento das Imagens

- f) Varie a dimensão do *viewport*.  
Verifique que a galeria tem capacidade para se ajustar às diferentes resoluções.
- g) A figura 4, na página seguinte, apresenta o aspeto final da galeria.

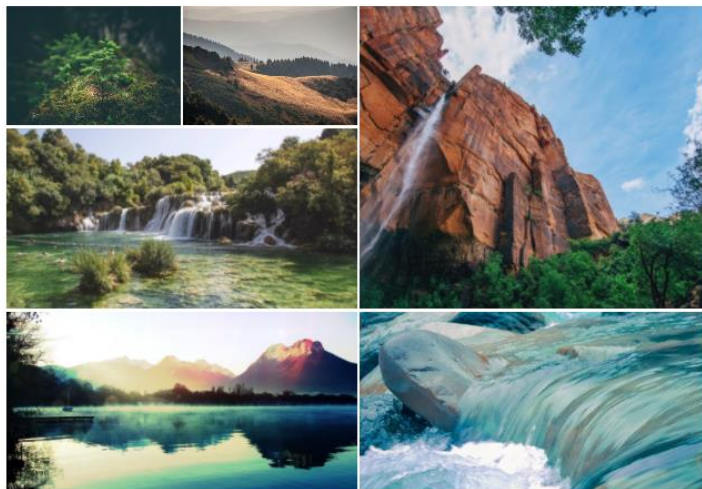


Figura 4 – Página com Galeria

### Exercício 3: CSS media queries – Galeria

- a) Apesar da galeria ter capacidade de se adaptar a diferentes dimensões do viewport, pretende-se definir uma nova organização de conteúdo para resoluções mais reduzidas. Assim, defina um *media query* para uma largura do *viewport* inferior a 575px e aplicada a todos os *media-type*.
- b) O elemento **div** cuja classe é **container** deve ocupar 96% da largura disponível do *viewport*.
- c) Implemente uma nova Grid ajustável em largura, apenas com uma coluna, com um gap correspondente a 1% da **largura** do *viewport*. A grid deve incluir 6 linhas com altura correspondente a 15% da **altura** do *viewport* (Figura 5)



Figura 5 – Grid para posicionamento das imagens nas resoluções mais baixas

- d) Redefina o posicionamento dos diversos itens de forma a que cada imagem ocupe uma célula da grid.
- e) A figura 4 apresenta o aspeto da galeria com *viewport* reduzido.

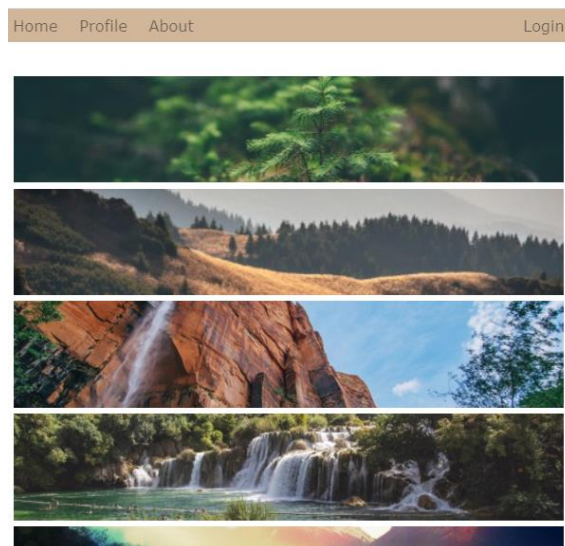


Figura 5 – Página com galeria com nas resoluções mais baixas

#### Exercício 4: CSS media queries – Menu

- a) Defina um CSS *media query* para uma largura de *viewport* inferior a 575px e aplicada a todos os *media-type* no ficheiro **ficha7\_Menu.css**.
- b) Reduza o tamanho da fonte (*font-size*) para 90% da dimensão por *default*.
- c) Especifique as regras para que o menu seja apresentado em coluna como apresentado na Figura 6. Pretende-se atingir o mesmo resultado recorrendo, no entanto, a duas abordagens alternativas (Flexbox/CSS Grid):
- Especifique as regras para quando o menu é implementado com recurso a flexbox. Para tal, pode aplicar as propriedades **flex-direction**, **text-align:center**;
  - Especifique as regras para quando o menu é implementado com recurso a CSS Grid. Pode recorrer às propriedades **grid-templates-columns**, **text-align:center**.
  - Como pode verificar, o último elemento (Login) fica encostado à direita em qualquer uma das abordagens, uma vez que ainda está a assumir a propriedade *margin-left:auto* implementada no exercício 1. Assim, **crie uma regra**, a ser aplicada ao último elemento (seja para a abordagem flex ou grid), de forma a que a propriedade *margin-left* seja 0.

**Nota:** Deve efetuar as alterações na estrutura HTML de forma a permitir que seja aplicada a formatação baseada em Flexbox / CSS grid.

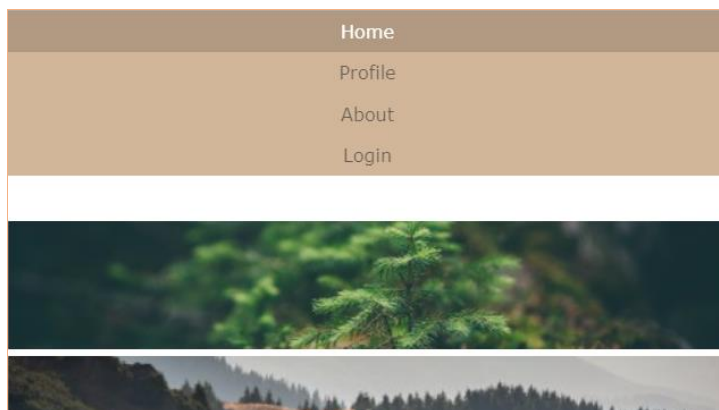


Figura 6 – Menu com nas resoluções mais baixas

### Exercício 5: Sandwich Menu

- a) Elimine os comentários do código HTML que engloba os elementos **label** e **input**. Estes permitirão implementar o *sandwich menu* como apresentado na figura 7.

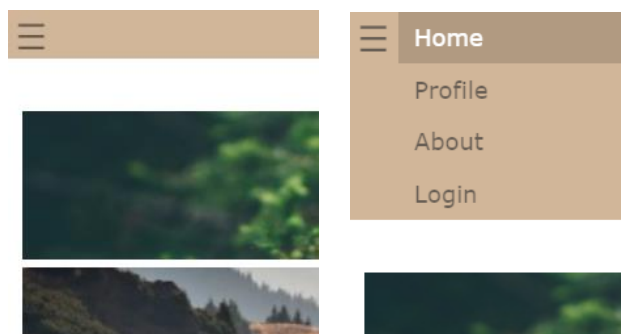


Figura 7 – Sandwich Menu encostado à esquerda

- b) Especifique uma regra para o elemento **label** de forma a que este não seja apresentado na janela do browser (`display:none;`), o tamanho de letra (`font-size`) seja 22px e possua um espaçamento interno à esquerda de 5px.
- c) Especifique uma regra para o elemento com **id toggle** não seja apresentado na janela do browser.
- d) Aplique ao elemento **nav** a classe **sandwich**.
- e) Para uma dimensão do *viewport* inferior a 575px:
- Sempre que a *class sandwich* estiver aplicada, faça com que o elemento **label** seja apresentado (`display:block`).

- O *menu*, sempre que a classe **sandwich** estiver aplicada, deve estar oculto e não ter qualquer influência no layout.
- Especifique uma regra para que sejam aplicados espaçamentos exteriores (margin) ao **menu** quando a classe **sandwich** estiver aplicada. O espaçamento deverá ser -30px na margem superior e 30px na margem esquerda e texto deverá estar encostado à esquerda (text-align).
- Especifique a seguinte regra para colocar o menu visível/invisível quando se clicar no icon do **menu**. A aplicação da regra pressupõe o recurso ao *Flexbox*, ou seja assumindo que o atributo **class** do **menu** não possui o valor **grid**:

```
.sandwich #toggle:checked + .menu:not(.grid) {  
    display: flex;  
}
```

- Tendo em consideração a regra anterior, especifique a mesma regra para o caso baseado em CSS Grid.

#### Exercício 5: Sandwich Menu à direita (Opcional)

Implemente as regras de forma a que o menu sandwich esteja encostado à direita.

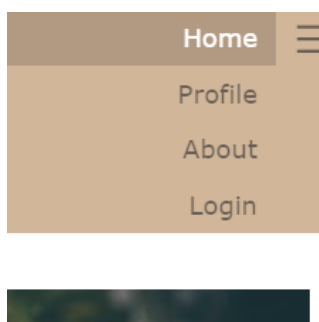


Figura 8 – Sandwich Menu encostado à direita