



Apoio

Curso: Técnico de Informática - Sistemas

**UFCD:** 0793 – Scripts CGI e folhas de estilos

Acção: Turma 1 - Técnico de Informática - Sistemas (EFA Técnico de Informática)

Formador: João Fragoso

**Data**: 22/09/2021

Nome do Formando/a:

## **CSS - Cascading Style Sheets**

Na utilização das CSS as propriedades podem ser declaradas de três maneiras diferentes.

A primeira delas é como um atributo style no próprio elemento:

```
0 conteúdo desta tag será exibido em azul com fundo amarelo no navegador!
```

**Desvantagem:** Temos que estar sempre a chamar o atributo style para cada Tag que queremos formatar

A outra maneira é declarar as propriedades dentro de uma tag <style>.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Sobre a Mirror Fashion</title>
    <style>
      p {
        background-color: yellow;
        color: blue;
      }
    </style>
  </head>
  <body>
     O conteúdo desta tag será exibido em azul com fundo amarelo!
    <strong>Também</strong> será exibido em azul com fundo amarelo!
    </body>
</html>
```











A terceira maneira de declararmos os estilos do nosso documento é com um arquivo externo, geralmente com a extensão .css.

```
<head>
  <meta charset="utf-8">
   <title>Sobre a Mirror Fashion</title>
  <link rel="stylesheet" href="estilos.css">
```

#### **Propriedades das Fontes**

A propriedade *font-family* pode receber o valor com aspas (nome da fonte) ou sem aspas (família da fonte).

Por padrão, os browsers mais conhecidos exibem num tipo que conhecemos como *serif* (família da fonte). A fonte mais conhecida é a "Times New Roman". São chamadas de **fontes serifadas** pelos pequenos ornamentos nas terminações.

Para definir em todo o documento inserir na tag body:

```
body {
  font-family: "Arial", "Helvetica", sans-serif;
}
```

O browser verifica se a fonte "Arial" está disponível aplica a todo o texto, caso a fonte "Arial" não esteja disponível, o browser verificará a disponibilidade da próxima fonte declarada, neste caso a "Helvetica". Caso o browser também não encontre essa fonte, ele solicita qualquer fonte que pertença à família "sans-serif".

Propriedade *font-style*: define o estilo da fonte podendo ser normal (normal na vertical), italic (inclinada) e oblique (oblíqua).

Outras propriedades:











#### **Padding**

A propriedade *padding* é utilizada para definir uma margem interna em alguns elementos (por margem interna queremos dizer a distância entre o limite do elemento e respetivo conteúdo) e tem as subpropriedades listadas a seguir:

- padding-top
- padding-right
- padding-bottom
- padding-left

Se for passado apenas um valor para a propriedade padding, esse valor é aplicado em todas as direções.

```
p {
  padding: 10px;
}
```

Se forem passados dois valores, o primeiro será aplicado acima e abaixo (equivalente a passar o mesmo valor para padding-top e padding-bottom) e o segundo será aplicado à direita e à esquerda (equivalente ao mesmo valor para padding-right e padding-left).

```
p {
   padding: 10px 15px;
}
```

Se forem passados três valores, o primeiro será aplicado acima (equivalente a paddingtop), o segundo será aplicado à direita e à esquerda (equivalente a passar o mesmo valor para padding-right e padding-left) e o terceiro valor será aplicado abaixo do elemento (equivalente a padding-bottom)

```
p {
  padding: 10px 20px 15px;
}
```

Se passados quatro valores, serão aplicados respectivamente a padding-top, padding-right, padding-bottom e padding-left. Para facilitar a memorização dessa ordem, basta lembrar que os valores são aplicados em **sentido horário**.

```
p {
  padding: 10px 20px 15px 5px;
}
```











#### Margin

A propriedade margin é semelhante à propriedade padding, só que vai adicionar espaço após o limite do elemento, ou seja, é um espaçamento além do elemento em si.

- margin-top
- margin-right
- margin-bottom
- margin-left

### Centrar Elementos na Página

Ainda sobre as classes, pode ser interessante criar uma classe que determina a centralização horizontal de qualquer elemento, sem interferir nas suas margens superior e inferior:

```
.container {
  margin-right: auto;
  margin-left: auto;
}
```

# Position (posicionamento dos elementos)

**Static:** Valor padrão para todos os elementos. Um elemento com posição static permanece sempre no seu local original no documento.

**Relative:** Aceita coordenadas. As coordenadas que passamos são em relação à posição original do elemento. Por exemplo:

```
.logotipo {
  position: relative;
  top: 20px;
  left: 50px;
}
```

Terá 20px adicionados ao seu topo e 50px adicionados à sua esquerda em relação à sua posição original. Estamos a adicionar pixeis de distância naquela direção, então o elemento **ficará mais abaixo e à direita** em comparação à sua posição original.











#### **Outro Exemplo:**

```
.relative1 {
  position: relative;
}
.relative2 {
  position: relative;
  top: 20px;
  loft: 20px;
  background-color: white;
  width: 500px;
}
```

```
Atribuindo as propriedades sop, rages, bettus, e tarté de um elemento retalivande de sua posição normal. Outro conteúdo não será ajustado para caber dentro de algum espaço que sobre do elemento.
```

**Absolute**: O elemento que tem o modo de posicionamento absolute toma como referência qualquer elemento que seja seu pai na estrutura do HTML cujo modo de posicionamento seja **diferente de static** (que é o padrão), e obedece às coordenadas de acordo com o tamanho total desse elemento pai.

```
.relative {
  position: relative;
  width: 600px;
  height: 400px;
}
.absolute {
  position: absolute;
  top: 120px;
  right: 0;
  width: 300px;
  height: 200px;
}
```

```
Este elemento está posicionado relativamente. Se este elemento tivesse posicionado relativamente ina escapar e ficaria posicionado relativo ao corpo do documento.

| Solve |
```





