

Desenvolvimento Web

Aula 4

Prof. Calebe Conceição

O que vimos na última aula?

Aprendemos os primeiros passos para desenvolver nossas páginas web

Conhecemos os ambientes de desenvolvimento

Construímos nossas primeiras páginas com HTML

Entendemos o que é o DOM

Realizamos práticas simples e enviamos para o servidor, para acesso público

Listas, Imagens, Tabelas, Formulários

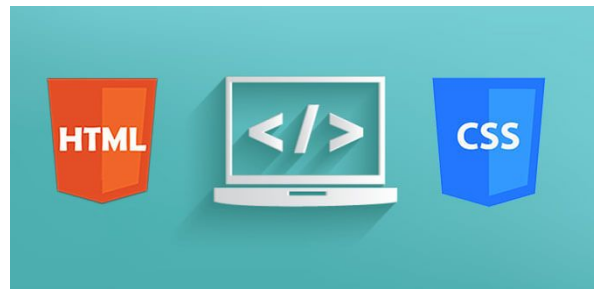
O que vamos ver hoje:

As páginas que construímos até aqui eram funcionais, mas muito feias!

Hoje vamos começar a aprender a definir estilos para elas!

Na nossa primeira aula...

- [Navegadores] Interpretam o código que vem em arquivos de um servidor
 - ou arquivo do PC mesmo
 - Front-end
 - **HTML**
 - **CSS**
 - *JavaScript*



Antes do CSS tudo era mato

- HTML é bem limitado
 - Como vocês podem ter notado, as páginas que vocês criaram até agora são bem simples
 - Sem muita possibilidade de formatação e organização do conteúdo



Da onde vem o CSS?

- Criado em 1994
 - Håkon Wium Lie
 - Trabalhava no CERN com o criador do HTML
 - Lançada mesmo em 1996
 - Suporte no IE3



Da onde vem o CSS?

- CSS está na versão 3
 - Já passou por alterações e evoluções
 - Mantido pela W3C
 - Suportado pela grande maioria dos navegadores
 - Facilita muito a vida no geral



HTML VS HTML + CSS



Esquina do Sabor

Promção do Dia!

Por tempo limitado, apresentamos a você a melhor promoção do dia!

Apresentamos a você a melhor promoção do dia!

Nosso Cardápio



Pastel de Queijo

Tradicional pastelinho frito por dentro. R\$ 7,00



Coxinha de Frango

Tradicional coxinha frita com molho de tomate. R\$ 5,00



Caju de Caju

Tradicional caju de caju com leite de coco. R\$ 5,00

Faça seu Pedido

Nome:

Endereço:

Telefone:

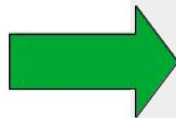
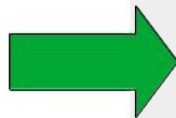
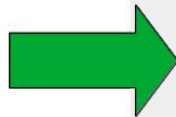
Quantidade:

Preço unitário:

Total:

Observações:

Enviar Pedido



Promção do Dia!

Por tempo limitado, apresentamos a você a melhor promoção do dia!

Apresentamos a você a melhor promoção do dia!

★ Nosso Cardápio



Pastel de Queijo

Tradicional pastelinho frito por dentro. R\$ 7,00

R\$ 7,00



Coxinha de Frango

Tradicional coxinha frita com molho de tomate. R\$ 5,00

R\$ 5,00



Caju de Caju

Tradicional caju de caju com leite de coco. R\$ 5,00

R\$ 5,00

Faça seu Pedido

Seus Dados

Nome:

Endereço:

Telefone:

Apresentamos a você a melhor promoção do dia!

Seu Pedido

Selecione o item:

Pastel de Queijo

Quantidade:

Preço unitário:

Total:

Observações:

Enviar Pedido

© 2023 Esquina do Sabor. Todos os direitos reservados.

Arquiteto: Designer

O que é o CSS?

➤ Cascading Style Sheets

- Folhas de estilo em cascata
- Não é linguagem de programação!
- É um linguagem de estilo para adicionar **estilos** ao HTML
 - fontes, cores, espaçamentos, bordas, etc
 - tudo que pode ser alterado no html de estilo
 - vários atributos podem ser alterados



O que é o CSS?

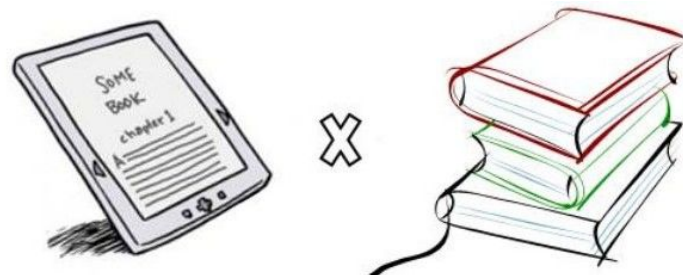
- Separar o estilo do conteúdo

- O conteúdo é o HTML

- O estilo é o CSS

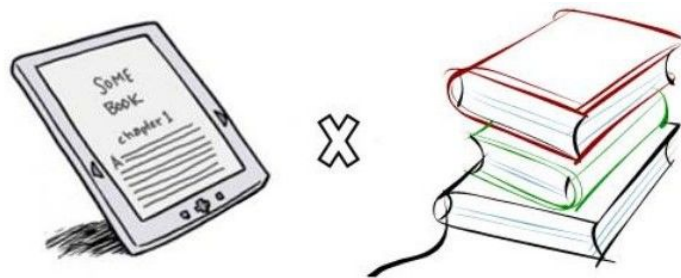
- A forma de apresentar pode variar de acordo com o estilo que daremos ao conteúdo

- Por que vocês acham que isso seria bom?



O que é o CSS?

- “Um código é escrito uma vez, e lido muitas vezes” *[alguém muito inteligente]*
- Organizar melhor o código
 - Isolar interesses
 - Estilo é estilo, e conteúdo é conteúdo!
 - Tornar o código mais fácil de ser
 - entendido
 - alterado



Como colocar CSS

- Existem 3 formas de inserir CSS no código HTML
 - inline ⇒ em linha
 - internal ⇒ interna
 - external ⇒ externa

CSS inline

➤ Testem aí...

```
<p style="color:#ff0000;  
font-size:48px;  
font-weight:bold;  
background-color:#0000ff;  
width:300px;"> Este é um  
parágrafo! </p>
```

```
<p style="color:#ff0000;  
font-size:48px;  
font-weight:bold;  
background-color:#0000ff;  
width:300px;"> Este é o  
segundo parágrafo! </p>
```

CSS inline

- Testem aí...
 - Qual o problema dessa forma de usar o css?

```
<p style="color:#ff0000;  
font-size:48px;  
font-weight:bold;  
background-color:#0000ff;  
width:300px;"> Este é um  
parágrafo! </p>
```

```
<p style="color:#ff0000;  
font-size:48px;  
font-weight:bold;  
background-color:#0000ff;  
width:300px;"> Este é o  
segundo parágrafo! </p>
```

CSS inline

- **Dentro do próprio marcador HTML**
 - **Somente aplicável para 1 marcador HTML**
- Não é recomendável de usar
 - Replicação de código
 - Mistura o código de conteúdo com o código de estilo
 - Difícil de achar dentro do html

```
<p style="color:#ff0000;  
font-size:48px;  
font-weight:bold;  
background-color:#0000ff;  
width:300px;"> Este é um  
parágrafo! </p>
```

```
<p style="color:#ff0000;  
font-size:48px;  
font-weight:bold;  
background-color:#0000ff;  
width:300px;"> Este é o  
segundo parágrafo! </p>
```

CSS interno

- **Dentro do próprio arquivos HTML**

- Separado dentro do `<head> </head>`
- Dentro do marcador `<style> </style>`
- Reduz o código replicado

- [ainda] Não é recomendável de usar

- Tem alguns problemas...

...

```
<head>
```

```
    <style>
```

```
        /*estilos
```

```
aqui!*/
```

```
    </style>
```

```
</head>
```

```
<body>
```

...

Como colocar CSS?

- Agora o estilo não está mais direto no marcador que deve ser customizado
 - Usando CSS interno, temos de identificar quais elementos
- Para isso o CSS tem 3 elementos básicos:

```
seletor { atributo : valor-do-atributo; }
```

Seletor

- Seleciona os marcadores html que serão afetados pelo estilo definido
 - sempre vem primeiro, antes da definição de estilo
 - uma das formas de seleção é colocar o marcador html que vai ser aplicado o estilo

seletor { atributo : valor-do-atributo; }

Separadores

- `{ }` ⇒ separador de bloco
 - define o estilo do seletor anterior ao `{ }`
- `:` ⇒ divide o atributo do valor
- `;` ⇒ encerra um par atributo-valor
 - podemos ter mais de um atributo por bloco

```
seletor {  
    atributo1 : valor-do-atributo1;  
    atributo2 : valor-do-atributo2;  
}
```

Atributo

- É a propriedade do HTML que vai ser modificada
- HTML tem muitas propriedades
 - Como vocês viram no exemplo do parágrafo anterior...

```
seletor { atributo : valor-do-atributo; }
```

Valor do atributo

- É o valor da propriedade que vai ser aplicado nos elementos css selecionados
- Altera o valor do atributo
 - de acordo com o atributo

```
seletor { atributo : valor-do-atributo; }
```

CSS inline

- Qual é o seletor que vamos usar?
- Quais são os atributos?
- Quais os valores?

```
<p style="color:#ff0000;  
font-size:48px;  
font-weight:bold;  
background-color:#0000ff;  
width:300px;"> Este é um  
parágrafo! </p>
```

```
<p style="color:#ff0000;  
font-size:48px;  
font-weight:bold;  
background-color:#0000ff;  
width:300px;"> Este é o  
segundo parágrafo! </p>
```

CSS inline

- Qual é o **seletor** que vamos usar?
- Quais são os **atributos**?
- Quais os **valores**?
- Como que ficaria o css interno disso?

```
<p style="color:#ff0000;  
font-size:48px;  
font-weight:bold;  
background-color:#0000ff;  
width:300px;"> Este é um  
parágrafo! </p>
```

```
<p style="color:#ff0000;  
font-size:48px;  
font-weight:bold;  
background-color:#0000ff;  
width:300px;"> Este é o  
segundo parágrafo! </p>
```

CSS interno

- Como que ficaria o CSS interno?

```
<!DOCTYPE html>
<html>
  <head> <title>Primeiro CSS </title>
    <style>
      p { color:#ff0000;
          font-size:48px;
          font-weight:bold;
          background-color:#0000ff;
          width:300px; }
    </style>
  </head>
  <body>
    <p> Este é um parágrafo! </p>
    <p> Este é o segundo parágrafo! </p>
  </body>
</html>
```


CSS interno

- Como que ficaria o CSS interno?
- O que acontece com o estilo dos p?

```
<!DOCTYPE html>
<html>
  <head> <title>Primeiro CSS </title>
    <style>
      p { color:#ff0000;
          font-size:48px;
          font-weight:bold;
          background-color:#0000ff;
          width:300px; }
    </style>
  </head>
  <body>
    <p> Este é um parágrafo! </p>
    <p> Este é o segundo parágrafo! </p>
  </body>
</html>
```

CSS interno

- Como que ficaria o CSS interno?
 - troca o estilo de todos os marcadores de combinam com o seletor!
 - todos os **p**, no caso...

```
<!DOCTYPE html>
<html>
  <head> <title>Primeiro CSS </title>
    <style>
      p { color:#ff0000;
          font-size:48px;
          font-weight:bold;
          background-color:#0000ff;
          width:300px; }
    </style>
  </head>
  <body>
    <p> Este é um parágrafo! </p>
    <p> Este é o segundo parágrafo! </p>
  </body>
</html>
```

CSS interno

- Troquem algum dos valores dos atributos para ver...

```
<!DOCTYPE html>
<html>
  <head> <title>Primeiro CSS </title>
    <style>
      p { color:#ff0000;
          font-size:48px;
          font-weight:bold;
          background-color:#0000ff;
          width:300px; }
    </style>
  </head>
  <body>
    <p> Este é um parágrafo! </p>
    <p> Este é o segundo parágrafo! </p>
  </body>
</html>
```

CSS interno

- Qual problema de usar css interno?
 - Reaproveitamento entre arquivos
 - Usar em mais de um arquivo HTML tem de copiar ele no head de cada um deles

```
<!DOCTYPE html>
<html>
  <head> <title>Primeiro CSS </title>
    <style>
      p { color:#ff0000;
          font-size:48px;
          font-weight:bold;
          background-color:#0000ff;
          width:300px; }
    </style>
  </head>
  <body>
    <p> Este é um parágrafo! </p>
    <p> Este é o segundo parágrafo! </p>
  </body>
</html>
```

CSS Externo

- Criar uma pasta /css
 - criar um arquivo **primeiro-estilo.css**
 - salvar o conteúdo do estilo do slide anterior

```
p {  
    color:#ff0000;  
    font-size:48px;  
    font-weight:bold;  
    background-color:#0000ff;  
    width:300px;  
}
```

CSS Externo

- Colocar o css dentro do HTML por link de estilo
 - `<link>`
 - `rel` ⇒ relationship
 - `href` ⇒ caminho para o css
- Criar o arquivo HTML ao lado (**css1.html**)

```
<!DOCTYPE html>
<html>
  <head> <title>Primeiro CSS </title>
    <link rel="stylesheet"
      href="css/primeiro-estilo.css">
  </head>
  <body>
    <p> Este é um parágrafo! </p>
    <p> Este é o segundo parágrafo! </p>
  </body>
</html>
```

CSS Externo

- Criar um arquivo novo html (**css1.html**)
 - Copiar o conteúdo do arquivo HTML (**css2.html**)
- Abram os dois arquivos html...

```
<!DOCTYPE html>
<html>
  <head> <title>Segundo CSS </title>
    <link rel="stylesheet"
      href="css/primeiro-estilo.css">
  </head>
  <body>
    <p> Este é um parágrafo do segundo
CSS!</p>
    <p> Este é o segundo parágrafo do
segundo CSS! </p>
  </body>
</html>
```

Seletores CSS

Seletores

- Para começar a aula
 - Criem um documento HTML
 - Criem um documento CSS
 - Criem o link (<link ... >)

Seletores

<p>Primeiro parágrafo.</p>

<p>Segundo parágrafo.</p>

<p>Terceiro parágrafo.</p>

- Eu tenho um texto
 - Quero aplicar um estilo só no primeiro parágrafo...
 - Deixar só o primeiro parágrafo em itálico...
 - Como que eu faço isso?

Seletores

```
p {  
    font-style: italic;  
}
```

- Se eu colocar esse estilo
 - Fica certo?
 - Fica só o primeiro parágrafo com estilo?

Seletores

`<p>Primeiro parágrafo.</p>`

`<p>Segundo parágrafo.</p>`

`<p>Terceiro parágrafo.</p>`

- Escolher um elemento específico para aplicar o estilo
 - Primeira passo é identificar esse elemento no HTML
 - **Como que eu faço isso?**

Atributo id

- Colocar um **id** no marcador html que queremos aplicar o estilo
 - id = "nomeDoElemento"
 - No caso do exemplo
 - Vamos colocar o id no parágrafo que queremos

```
<p id="primeiroParagrafo">
```

```
Primeiro parágrafo.</p>
```

```
<p>Segundo parágrafo.</p>
```

```
<p>Terceiro parágrafo.</p>
```

Seletor de id

```
p {  
    font-style: italic;  
}
```

- Segundo passo é colocar o estilo para o id no css
 - Como que faz isso?

Seletor de id

```
#primeiroParagrafo {  
    font-style: italic;  
}
```

- Ao invés de colocar o marcador
- Colocar o próprio **id** do marcador, mas com **#** na frente
 - **#idNoHtml**
 - No nosso exemplo
 - **#primeiroParagrafo**

Seletor de id

➤ Então...

- Para colocar um estilo usando ids é preciso
 - 1) Definir o estilo no css
 - Utilizando #
 - Escolher o nome do id a ser utilizado
 - Escrever o estilo dentro do {}
 - 2) Colocar o id no marcador html
 - id="nomeDoid"

```
#primeiroParagrafo {  
    font-style: italic;  
}
```


Seletor de id

```
#primeiroParagrafo {  
    font-style: italic;  
}
```

- Da para colocar vários ids no css e no html?

Seletor de id

- Da para colocar vários ids no css e no html?
 - Dá!
 - Copiem os estilos e façam os estilos funcionarem no html de você

```
#primeiroParagrafo {  
    font-style: italic;  
}
```

```
#segundoParagrafo {  
    font-weight: bold;  
}
```

```
#terceiroParagrafo {  
    font-size: 32px;  
}
```

Seletor de id

- Se eu quiser agora colocar a mesma fonte para todos os parágrafos...
 - Como eu faço?

```
#primeiroParagrafo {  
    font-style: italic;  
}
```

```
#segundoParagrafo {  
    font-weight: bold;  
}
```

```
#terceiroParagrafo {  
    font-size: 32px;  
}
```

Seletor de id

- Agora cada parágrafo tem um estilo diferente
- **O que acontece se eu colocar esse estilo no marcador p?**
 - sem nenhum id selecionado...

```
p {  
    font-family: Helvetica;  
}  
#primeiroParagrafo {  
    font-style: italic;  
}  
#segundoParagrafo {  
    font-weight: bold;  
}  
#terceiroParagrafo {  
    font-size: 32px;  
}
```

Seletor de id

- Dessa forma todos os parágrafos vão ter a fonte Helvetica
 - E também vão ter o estilo específico do id deles

```
p {  
    font-family: Helvetica;  
}  
#primeiroParagrafo {  
    font-style: italic;  
}  
#segundoParagrafo {  
    font-weight: bold;  
}  
#terceiroParagrafo {  
    font-size: 32px;  
}
```

Seletor de id

- E se eu resolver colocar uma outra fonte em um dos estilo com id
 - Qual fonte que fica?
 - Testem aí...

```
p {  
    font-family: Helvetica;  
}  
  
#primeiroParagrafo {  
    font-style: italic;  
    font-family: Times;  
}  
  
#segundoParagrafo {  
    font-weight: bold;  
}  
  
#terceiroParagrafo {  
    font-size: 32px;  
}
```

Seletor de id

- E se eu resolver colocar uma outra fonte em SOMENTE UM dos parágrafos com id
 - Como que eu faço?

```
p {  
    font-family: Helvetica;  
}  
#primeiroParagrafo {  
    font-style: italic;  
}  
#segundoParagrafo {  
    font-weight: bold;  
}  
#terceiroParagrafo {  
    font-size: 32px;  
}
```

Seletor de id

- De novo...
 - O estilo mais específico que prevalece
 - A prioridade é:
 - Inline > ID > Classe > Tag
 - id é mais específico do que marcador
 - **Logo, ficamos com o font-family do id**

```
p {  
    font-family: Helvetica;  
}  
#primeiroParagrafo {  
    font-style: italic;  
    font-family: Times;  
}  
#segundoParagrafo {  
    font-weight: bold;  
}  
#terceiroParagrafo {  
    font-size: 32px;  
}
```


Seletores

- Quero aplicar um estilo no h1 e no primeiro parágrafo
- **Como fazer?**

```
<h1> Título bonito </h1>
```

```
<p id="primeiroParagrafo">
```

```
Primeiro parágrafo.</p>
```

```
<p id="segundoParagrafo">
```

```
Segundo parágrafo.</p>
```

```
<p id="terceiroParagrafo">
```

```
Terceiro parágrafo.</p>
```

Seletores

- Colocar os elementos a serem selecionados entre vírgulas
 - **elemento,elemento,...**
 - pode ter quantos quiser na sequência
 - Ou seja, ele concatena seletores e o estilo é aplicado em todos os selecionados

```
h1, #primeiroParagrafo {  
    font-family: Helvetica;  
}  
  
#primeiroParagrafo {  
    font-style: italic;  
}  
  
#segundoParagrafo {  
    font-weight: bold;  
}  
  
#terceiroParagrafo {  
    font-size: 32px;  
}
```

Seletores

- **O que seleciona o seletor abaixo?**
 - `h1,h2,#primeiroParagrafo, span,div`

```
h1,#primeiroParagrafo {  
    font-family: Helvetica;  
}  
  
#primeiroParagrafo {  
    font-style: italic;  
}  
  
#segundoParagrafo {  
    font-weight: bold;  
}  
  
#terceiroParagrafo {  
    font-size: 32px;  
}
```

Seletores

- Alterem o HTML de vocês

```
<h1> Título bonito </h1>
```

```
<p id="primeiroParagrafo">
```

```
Primeiro parágrafo tem uma  
parte em <span> destaque  
</span>.</p>
```

```
<p id="segundoParagrafo">
```

```
Segundo parágrafo <span>  
texto aleatório</span>.</p>
```

```
<p id="terceiroParagrafo">
```

```
Terceiro parágrafo.</p>
```

Seletores

- Como que fazemos para aplicar um estilo somente ao span do primeiro parágrafo?

```
<h1> Título bonito </h1>
```

```
<p id="primeiroParagrafo">
```

```
Primeiro parágrafo tem uma  
parte em <span> destaque
```

```
</span>.</p>
```

```
<p id="segundoParagrafo">
```

```
Segundo parágrafo <span>
```

```
texto aleatório</span>.</p>
```

```
<p id="terceiroParagrafo">
```

```
Terceiro parágrafo.</p>
```

Seletores

➤ Selecionar com base na hierarquia do documento

- `#primeiroParagrafo span`
- seleciona todos os spans dentro do `#primeiroParagrafo`
- Coloquem outro span dentro do `#primeiroParagrafo`

```
h1 {  
    font-family: Helvetica;  
}  
#primeiroParagrafo {  
    font-style: italic;  
}  
#primeiroParagrafo span {  
    text-transform:  
uppercase;  
}  
#segundoParagrafo {  
    font-weight: bold;  
}  
#terceiroParagrafo {  
    font-size: 32px;
```

Seletores

➤ Dá para colocar vários termos da hierarquia

- `div p span a`
- Vai seleccionar o que?

```
h1 {  
    font-family: Helvetica;  
}  
#primeiroParagrafo {  
    font-style: italic;  
}  
#primeiroParagrafo span {  
    text-transform:  
uppercase;  
}  
#segundoParagrafo {  
    font-weight: bold;  
}  
#terceiroParagrafo {  
    font-size: 32px;
```

Seletores

➤ **Dá para usar tanto ids quanto marcadores**

- seletores selecionam os elementos do HTML!
- São formas de escolher o que vamos aplicar o estilo que tá dentro daquele seletor

```
h1 {  
    font-family: Helvetica;  
}  
#primeiroParagrafo {  
    font-style: italic;  
}  
#primeiroParagrafo span {  
    text-transform:  
uppercase;  
}  
#segundoParagrafo {  
    font-weight: bold;  
}  
#terceiroParagrafo {  
    font-size: 32px;
```


Seletor de classe

Seletor de classe

- Citem tudo novo (ou limpem os anteriores)
 - Criem um documento HTML
 - Criem um documento CSS
 - Criem o link (<link ... >)

HTML

- Coloquem o código no arquivo HTML de vocês

<p>Error 01: Código não compilou na linha 10.</p>

<p>Warning 01: Código duplicado nas linhas 5-8 e 13-16.</p>

<p>Error 02: Tipos não compatíveis na linha 3.</p>

<p>Error 03: Tipo não definido.</p>

<p>Success 01: Compilado com sucesso.</p>

<p>Warning 02: Código nunca utilizado nas linhas 20-30.</p>

<p>Success 02: Compilado com sucesso.</p>

HTML

- O que temos de alterar para que fique assim?
- Façam aí um estilo de cada
 - Error
 - Success
 - Warning

Error 01: Código não compilou na linha 10.

Warning 01: Código duplicado nas linhas 5-8 e 13-16.

Error 02: Tipos não compatíveis na linha 3.

Error 03: Tipo não definido.

Success 01: Compilado com sucesso.

Warning 02: Código nunca utilizado nas linhas 20-30.

Success 02: Compilado com sucesso.

HTML

- O que temos de alterar para que fique assim?
 - Dá para fazer usando id?

Error 01: Código não compilou na linha 10.

Warning 01: Código duplicado nas linhas 5-8 e 13-16.

Error 02: Tipos não compatíveis na linha 3.

Error 03: Tipo não definido.

Success 01: Compilado com sucesso.

Warning 02: Código nunca utilizado nas linhas 20-30.

Success 02: Compilado com sucesso.

HTML

- O que temos de alterar para que fique assim?
 - Dá para fazer usando id?
 - Da...
 - Só vai dar um certo trabalho
 - **Qual o problema de usar ids aqui?**

Error 01: Código não compilou na linha 10.

Warning 01: Código duplicado nas linhas 5-8 e 13-16.

Error 02: Tipos não compatíveis na linha 3.

Error 03: Tipo não definido.

Success 01: Compilado com sucesso.

Warning 02: Código nunca utilizado nas linhas 20-30.

Success 02: Compilado com sucesso.

HTML

- Alterem o código no arquivo HTML de vocês

`<p>Error 01:`

`Código não compilou na linha 10. </p>`

`<p>Warning`

`01: Código duplicado nas linhas 5-8 e 13-16. </p>`

`<p>Error 02:`

`Tipos não compatíveis na linha 3. </p>`

`<p>Error 03:`

`Tipo não definido. </p>`

`<p>Success`

`01: Compilado com sucesso. </p>`

`<p>Warning`

`02: Código nunca utilizado nas linhas 20-30.</p>`

`<p> Success`

`02: Compilado com sucesso. </p>`

CSS

- Coloquem o código no arquivo CSS de vocês

```
body { font-family: "Courier new"; }  
.error{  
    font-size: 140%;  
    font-weight: bold;  
    color: red;  
}  
.warning {  
    font-size: 140%;  
    font-weight: bold;  
    color: #CCCC00;  
}  
.success {  
    font-size: 140%;  
    font-weight: bold;  
    color: darkGreen;  
}
```


Seletor de classe

- valor no atributo classe do html
- parecido com o id
- no css é o .
 - seleciona todos os elementos do html que tenham aquela classe

```
body { font-family: "Courier new"; }  
.error{  
    font-size: 140%;  
    font-weight: bold;  
    color: red;  
}  
.warning {  
    font-size: 140%;  
    font-weight: bold;  
    color: #CCCC00;  
}  
.success {  
    font-size: 140%;  
    font-weight: bold;  
    color: darkGreen;  
}
```

Seletor de classe

- Ainda dá para melhorar um pouco mais isso aí
- **O que dá para fazer com esse estilo para melhorar?**

```
body { font-family: "Courier new"; }  
.error{  
    font-size: 140%;  
    font-weight: bold;  
    color: red;  
}  
.warning {  
    font-size: 140%;  
    font-weight: bold;  
    color: #CCCC00;  
}  
.success {  
    font-size: 140%;  
    font-weight: bold;  
    color: darkGreen;  
}
```

CSS

- Coloquem o código no arquivo CSS de vocês
- E o html?

```
body {  
    font-family: "Courier new";  
}  
  
.status{  
    font-size: 140%;  
    font-weight: bold;  
}  
  
.error{  
    color: red;  
}  
  
.warning {  
    color: #CCCC00;  
}  
  
.success {  
    color: darkGreen;  
}
```

HTML

- Alterem o código no arquivo HTML de vocês

```
<p><span class="status error">Error
01:</span> Código não compilou na linha 10.
</p>
<p><span class="status warning">Warning
01:</span> Código duplicado nas linhas 5-8 e
13-16. </p>
<p><span class="status error">Error
02:</span> Tipos não compatíveis na linha 3.
</p>
<p><span class="status error">Error
03:</span> Tipo não definido. </p>
<p><span class="status success">Success
01:</span> Compilado com sucesso. </p>
<p><span class="status warning">Warning
02:</span> Código nunca utilizado nas linhas
20-30.</p>
<p> <span class="status success"> Success
02:</span> Compilado com sucesso. </p>
```

CSS

- Qual a diferença entre usar id e usar class?

```
body {  
    font-family: "Courier new";  
}  
  
.status{  
    font-size: 140%;  
    font-weight: bold;  
}  
  
.error{  
    color: red;  
}  
  
.warning {  
    color: #CCCC00;  
}  
  
.success {  
    color: darkGreen;  
}
```

Id ou class?

- Qual a diferença entre usar id e usar class?
 - id é um por elemento
 - class a pode ter várias classes em um mesmo elemento

```
<p><span class="status error">Error
01:</span> Código não compilou na linha 10.
</p>
```

```
<p><span class="status warning">Warning
01:</span> Código duplicado nas linhas 5-8 e
13-16. </p>
```

```
<p><span class="status error">Error
02:</span> Tipos não compatíveis na linha 3.
</p>
```

```
<p><span class="status error">Error
03:</span> Tipo não definido. </p>
```

```
<p><span class="status success">Success
01:</span> Compilado com sucesso. </p>
```

```
<p><span class="status warning">Warning
02:</span> Código nunca utilizado nas linhas
20-30.</p>
```

```
<p> <span class="status success"> Success
02:</span> Compilado com sucesso. </p>
```

Combinators

- Alguém sabe do que se trata?

Combinators

- Sim, existem mais seletores...
 - descendant ⇒ descendente
 - child ⇒ filhos diretos
 - adjacent sibling ⇒ irmão adjacente
 - general sibling selector ⇒ irmão qualquer

Static

- Coloquem aí no HTML
 - para que a gente consiga testar os combinators

```
<div>
```

```
  <p>parágrafo 1 tá no div de  
fora.</p>
```

```
  <p>parágrafo 2 tá no div de  
fora</p>
```

```
  <span>
```

```
    <p>parágrafo 3 tá no span  
    interno.</p>
```

```
  </span>
```

```
</div>
```

```
<p>parágrafo 4 não tá no div!</p>
```

```
<p>parágrafo 5 não tá no div!</p>
```

Descendant

```
div p {  
    background-color: black;  
    color: white;  
}
```

- Descendente
 - A gente já viu esse
 - Dado um seletor ele pega todos elementos que são descendentes do elemento
 - Coloque no CSS

Child

```
div > p {  
    background-color: black;  
    color: white;  
}
```

- Filhos diretos
 - Dado um seletor ele pega todos elementos que são descendentes diretos do elemento
 - Coloque no CSS

Adjacent Sibling

```
div + p {  
    background-color: black;  
    color: white;  
}
```

- Irmão adjacente
 - Pega um elemento que é diretamente adjacente ao seletor anterior

General Siblings

```
div ~ p {  
    background-color: black;  
    color: white;  
}
```

- Todos os irmãos adjacentes
 - Seleciona todos os irmãos que são daquele seletor

Cores

CSS Externo

- Alterar as cores do nosso site
- Aula passada vimos que é possível
 - Em algumas aulas de css
 - vimos que existe a forma de definir cores
 - alguém lembra?

CSS Externo

- Alterar as cores do nosso site
- Aula passada vimos que é possível
 - Em algumas aulas de css
 - vimos que existe a forma de definir cores
 - alguém lembra?
 - 6 dígitos hexadecimais (#ffffff)

#F0F8FF	
#FAEBD7	
#00FFFF	
#7FFFD4	
#F0FFFF	
#F5F5DC	
#FFE4C4	
#000000	
#FFEB3D	
#0000FF	
#8A2BE2	
#A52A2A	
#DEB887	
#5F9EA0	
#7FFF00	
#D2691E	
#FF7F50	
#6495ED	
#FFF8DC	
#DC143C	
#00FFFF	
#00008B	
#000000	

Valores de Cores

- Existem várias formas de definir cores
 - Hexadecimal
 - Nome
 - RGB(A)
 - HSL(A)
- Cada um deles produz as mesmas cores
 - Só que de formas diferentes
 - Os (A) no final são com transparências

Valores de Cores

- Composto por # e mais 6 dígitos hexadecimais (0-f)
 - $16^6 = 16.777.216$ de cores
 - Cada par de dígitos é uma cor do RGB
 - na ordem do RGB, ou seja
 - **#ff0000 é vermelho (red)**
 - **#00ff00 é verde (green)**
 - **#0000ff é azul (blue)**

Cores Nomes

- Nomes em Inglês de algumas cores
 - 140 cores, o nome da cor mesmo
 - blue, red, green, ...
 - variações (nem todas cores tem)
 - dark, light, medium, pale, ...
 - darkblue
 - lightblue
 - mediumblue

#F0F8FF	
#FAEBD7	
#00FFFF	
#7FFFD4	
#F0FFFF	
#F5F5DC	
#FFE4C4	
#000000	
#FFEB3D	
#0000FF	
#8A2BE2	
#A52A2A	
#DEB887	
#5F9EA0	
#7FFF00	
#D2691E	
#FF7F50	
#6495ED	
#FFF8DC	
#DC143C	
#00FFFF	
#00008B	
#000000	

Cores nomes

```
h1 {color: blue; }  
h2 {color: lightblue; }  
h3 {color: mediumblue;  
}  
h4 {color: darkblue; }
```

- Criem uma página html de exemplo
 - Com marcadores h1, h2, h3 e h4
 - Criem o arquivo css
 - Coloquem o arquivo css no html externamente
 - Coloquem no css o exemplo para ver

Cores RGB e HSL

- RGB é o hexadecimal disfarçado

- função `rgb(int,int,int)`
 - cada int vai de 0-255 (00-ff)
 - `rgb(255,0,0) ⇒ red`

- HSL é diferente

- função `hsl(int, %, %)`
- int = matiz ~ cor
- % = saturação
- % = brilho

```
h1 {color: hsl(240,  
100%, 50%); }
```

```
h2 {color:  
rgb(0,0,255); }
```

```
h3 {color: mediumblue;  
}
```

```
h4 {color: darkblue; }
```

Cores RGB e HSL

- RGBA e HSL são os mesmo rgb e hsl

- só que com alpha
- função `rgba(int,int,int,double)`
- função `hsla(int, %,%,double)`
 - o double é o alpha
 - quanto transparente é a cor

- testem aí

```
h1 {background-color:  
    rgba (0,0,255,0.9) ; }
```

```
h2 {background-color:  
    rgba (0,0,255,0.7) ; }
```

```
h3 {background-color:  
    rgba (0,0,255,0.5) ; }
```

```
h4 {background-color:  
    rgba (0,0,255,0.3) ; }
```

```
h5 {background-color:  
    rgba (0,0,255,0.1) ; }
```

Valores de Cores

- Normalmente
 - **Hexadecimal**
 - **Nome**
- Caso algum dia na vida de vocês precisem, vocês sabem o que são os dois
 - RGB(A)
 - HSL(A)

px

- O que é esse px?

```
p {  
    color : brown;  
    text-align: center;  
    line-height: 25px;  
    letter-spacing: 2px;  
    text-transform: uppercase;  
    word-spacing: 10px;  
    text-indent: 20px;  
    text-shadow: 3px 2px gray;  
}
```


px

- É uma das unidades de medida do CSS
 - **px** ⇒ **pixels**
 - Quais as outras?

```
p {  
    color : brown;  
    text-align: center;  
    line-height: 25px;  
    letter-spacing: 2px;  
    text-transform: uppercase;  
    word-spacing: 10px;  
    text-indent: 20px;  
    text-shadow: 3px 2px gray;  
}
```

Medidas Absolutas

- px ⇒ pixel
- mm ⇒ milímetro
 - 1mm ⇒ 3.78px
- in ⇒ polegada
 - 1in ⇒ 96px
- pt ⇒ pontos
 - 1pt ⇒ 1.33px
- pc ⇒ paicas
 - 1pc ⇒ 16px

```
p {  
    color : brown;  
    text-align: center;  
    line-height: 25px;  
    letter-spacing: 2px;  
    text-transform: uppercase;  
    word-spacing: 10px;  
    text-indent: 20px;  
    text-shadow: 3px 2px gray;  
}
```

Medidas Absolutas

- Pixel é a mais usada
- Da para usar qualquer uma
 - Coloquem outra unidade de medida para testar

```
p {  
    color : brown;  
    text-align: center;  
    line-height: 25px;  
    letter-spacing: 2px;  
    text-transform: uppercase;  
    word-spacing: 10px;  
    text-indent: 20px;  
    text-shadow: 3px 2px gray;  
}
```

Medidas Absolutas

- Pixel é a mais usada
- Da para usar qualquer uma
 - Coloquem outra unidade de medida para testar
 - **Qual motivos que pixel é a mais usada?**

```
p {  
    color : brown;  
    text-align: center;  
    line-height: 25px;  
    letter-spacing: 2px;  
    text-transform: uppercase;  
    word-spacing: 10px;  
    text-indent: 20px;  
    text-shadow: 3px 2px gray;  
}
```

Medidas Absolutas

- Pixel é a mais usada
- Da para usar qualquer uma
 - Coloquem outra unidade de medida para testar
 - Qual motivos que pixel é a mais usada?
 - **Pixel é relativo ao tamanho da tela**

```
p {  
    color : brown;  
    text-align: center;  
    line-height: 25px;  
    letter-spacing: 2px;  
    text-transform: uppercase;  
    word-spacing: 10px;  
    text-indent: 20px;  
    text-shadow: 3px 2px gray;  
}
```

Box model

Box model

- Alguém sabe o que é?

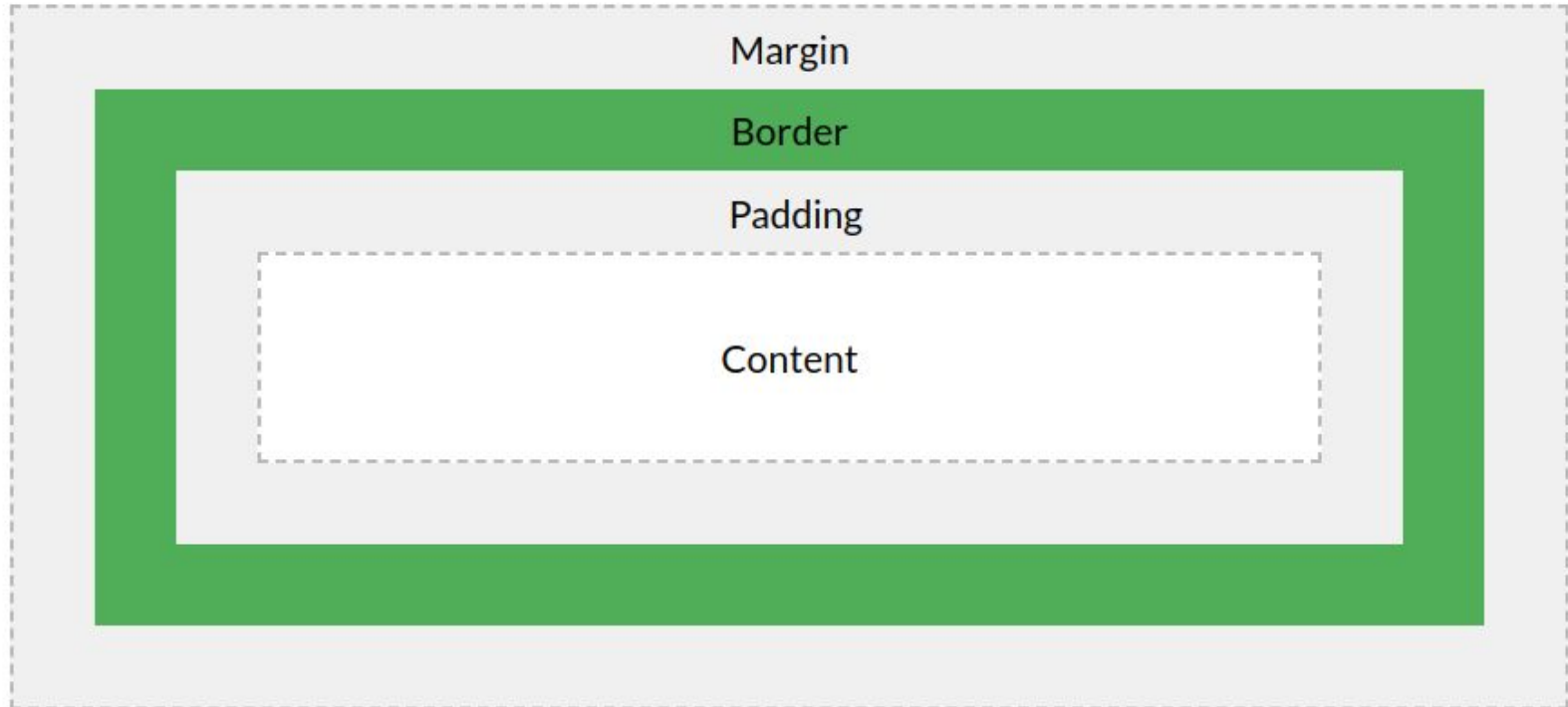
Box model

- Imagino que vocês já notaram no html que
 - alguns elementos tem mais espaçamento
 - outros têm menos

Box model

- É como o html lida com espaçamentos do conteúdo
- Envolve 3 propriedades que são importantes para o estilo
 - Margin
 - Border
 - Padding

Box model



Box model

- Peguem uma página qualquer
- Cliquem com o segundo botão em cima de algum elemento
 - Cliquem em inspecionar

Box model

- Peguem uma página qualquer
- Cliquem com o segundo botão em cima de algum elemento
 - Cliquem em inspecionar
 - Devem ter visto algo parecido com a figura anterior só que do objeto específico...

Border

Border

- Criem um arquivo html
 - Coloquem 11 parágrafos
- Criem um arquivo css
- Linkar o arquivo css ao html

Border

- Coloquem mais esse código CSS no arquivo CSS de vocês

```
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {
    border-style: dotted dashed solid
    double;}
```

Border

- Coloquem as classes uma em cada parágrafo para vocês verem como fica esse css

```
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {
    border-style: dotted dashed solid
    double;}
```


Border

- Cada uma delas é diferente
- A última é uma para cada borda
 - Sentido horário
 - top, right, bottom, left

```
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.none {border-style: none;}
p.hidden {border-style: hidden;}
p.mix {
    border-style: dotted dashed solid
    double;}
```

Border

- O que acontece alterando os estilos anteriores?

```
p.solid {  
    border-style: solid;  
    border-width: 5px;  
    border-color: red;  
}
```

```
p.dotted {  
    border-style: solid;  
    border-width: 10px;  
    border-color: red green yellow  
blue;  
}
```

Border

- Podemos também ter bordas individuais
 - Cada lado definido especificamente

```
p {  
    border-top-style: dotted;  
    border-right-style: solid;  
    border-bottom-style: dotted;  
    border-left-style: solid;  
}
```

Border

```
p.solid {  
    border-left: red 6px solid ;  
    background-color: lightgrey;  
}
```

- Podemos também ter bordas individuais

```
p.dotted {  
    border-left: dotted darkGreen 6px;  
    background-color: lightgrey;  
}
```

- borda individual a gente pode definir todos os estilos daquele lado numa linha só
- não precisa de ordem

Border

- Com base no que a gente viu até agora...
- Façam o html de vocês ficar assim ⇒

Error 01: Código não compilou na linha 10.

Warning 01: Código duplicado nas linhas 5-8 e 13-16.

Error 02: Tipos não compatíveis na linha 3.

Error 03: Tipo não definido.

Success 01: Compilado com sucesso.

Warning 02: Código nunca utilizado nas linhas 20-30.

Success 02: Compilado com sucesso.

Border

- Não é só um atributo de parágrafos!
- É aplicada a praticamente todos os elementos que a gente
- Criem um botão e apliquem a classe ao lado...

```
.botao {  
    text-decoration: none;  
    background-color: #4CAF50;  
    border: none;  
    color: white;  
    text-align: center;  
    font-size: 16px;  
}
```

Border-radius

- O que acontece aí se colocar esse border-radius?

```
.botao {  
    text-decoration: none;  
    background-color: #4CAF50;  
    border: none;  
    color: white;  
    text-align: center;  
    font-size: 16px;  
}
```

Border-radius

- Deixa os cantos mais redondinhos
- Quando maior o número mais redondo fica

```
.botao {  
    text-decoration: none;  
    background-color: #4CAF50;  
    border: none;  
    color: white;  
    text-align: center;  
    font-size: 16px;  
    border-radius: 5px;  
}
```


Margin

Margin

- O que acontece?

```
p {  
  border: 3px solid black;  
  margin-top: 100px;  
  margin-bottom: 100px;  
  margin-right: 150px;  
  margin-left: 80px;  
}
```

Margin

```
p {  
  border: 3px solid black;  
  margin: 25px 50px 75px 100px;  
}
```

➤ O que acontece?

Margin

```
p {  
    border: 3px solid black;  
    margin: 25px 50px 75px 100px;  
}
```

- Margem de cada lado
 - cada um com seu valor específico
 - sentido horário começando pelo top
 - top right bottom left

Margin

```
p {  
  border: 3px solid black;  
  margin: 50px;  
}
```

- O que acontece?

Margin

```
p {  
    border: 3px solid black;  
    margin: 50px;  
}
```

- Margem 50px em todos os lados
 - Não precisa definir todos os lados se eles forem iguais

Margin

```
p {  
  border: 3px solid black;  
  margin: auto;  
}
```

- O que acontece?

Margin

- Coloca o elemento no meio horizontalmente do elemento
- Se estiver dentro de um div?

```
div {  
    width: 500px;  
}  
p {  
    border: 3px solid black;  
    margin: auto;  
}
```


Margin

- Colapsar a margem
 - Muitos elementos têm margem por padrão
 - Para tirar a margem temos de definir para **margin: 0**

```
div {  
    width: 500px;  
}  
p {  
    border: 3px solid black;  
    margin: 0;  
}
```

Padding

Padding

- O que acontece?

```
p {  
  padding-top: 50px;  
  padding-right: 30px;  
  padding-bottom: 50px;  
  padding-left: 80px;  
}
```

Padding

```
p {  
    padding: 25px 50px 75px 100px;  
}
```

- O que acontece?

Padding

```
p {  
    padding: 30px;  
}
```

- O que acontece?

Padding

```
p {  
    padding: auto;  
}
```

- Basicamente são os mesmos valores do margin só que ao invés de ser o espaçamento externo é o interno

```
div {  
    padding: 0;  
}
```

MUITO
OBRIGADO