

INSTITUTO FEDERAL

Paraná

Campus Paranaguá

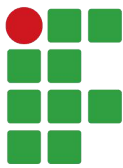
CSS

Desenvolvimento Web

Prof. Diego Stiehl

Nosso Site


- Já montamos o conteúdo do nosso site
 - Baixar HTML de exemplo no Moodle
- Colocamos as devidas marcações (significado)
 - Títulos, parágrafos, ênfases, estruturação da página, configurações, ...
- Criamos links
- Adicionamos imagens



INSTITUTO
FEDERAL

E a aparência?


Instituto Federal do Paraná

 INSTITUTO FEDERAL
Paraná

Campus Paranaguá

- [O Instituto](#)
- [Localização](#)
- [Quem Somos](#)
- [Cursos](#)
- [Estrutura](#)

O Instituto



O Instituto Federal do Paraná (IFPR) é uma instituição pública federal de ensino vinculada ao Ministério da Educação (MEC) por meio da Secretaria de Educação Profissional e Tecnológica (SETEC). É voltada a educação superior, básica e profissional, especializada na oferta gratuita de educação profissional e tecnológica nas diferentes modalidades e níveis de ensino.

A instituição foi criada em dezembro de 2008 através da Lei 11.892, que instituiu a Rede Federal de Educação Profissional e Tecnológica e os 38 institutos federais hoje existentes no país. Com a Lei em vigor, a Escola Técnica da Universidade Federal do Paraná (ET-UFPR) foi transformada no IFPR, que hoje possui autonomia administrativa e pedagógica.

O IFPR está presente nas cidades: Assis Chateaubriand, Campo Largo, Cascavel, Curitiba, Foz do Iguaçu, Irati, Ivaiporã, Jacarezinho, Londrina, Palmas, **Paranaguá**, Paranavai, Telêmaco Borba, Umuarama.

Também está em fase de implantação nas cidades: Pinhais, Pitanga, União da Vitória, Jaguaíva, Colombo, Capanema (3ª fase de ampliação da Rede Federal de Educação Profissional, Científica e Tecnológica).

Além dos câmpus, o IFPR está implantando as Unidades de Educação Profissional (UEP). São unidades especializadas na oferta de ensino técnico, cursos de formação inicial e continuada e de Educação a Distância, atendendo prioritariamente as ações que integram o Programa Nacional de Acesso ao Ensino Técnico e Emprego (Pronatec).

Vinculadas aos câmpus, as UEPs estarão presentes em oito municípios: Astorga, Goleerê, Quedas do Iguaçu, Coronel Vivida, Barracão, Bandeirantes, Guaira e Lapa.


Localização

Paranaguá significa "Grande Mar Redondo" (Pernagoa) em tupi guarani, sendo "Paraná" o significado de *Grande Rio* e "Goá" o significado de *Redondo*, uma referência à Baía de Paranaguá. É um município brasileiro localizado no litoral do estado do Paraná. É a cidade mais antiga do estado, sendo também chamada de "berço da civilização paranaense". A economia da cidade está intimamente ligada ao funcionamento do Porto Dom Pedro II, importante terminal, corredor de exportação do Estado, além do comércio, turismo, agricultura e pesca.

O Campus Paranaguá tem sua proposta sustentada nos arranjos produtivos locais e está localizado no seguinte endereço:

Rua : Antônio Carlos Rodrigues, 453. Bairro : Porto Seguro – Paranaguá – Paraná Cep: 83215-750

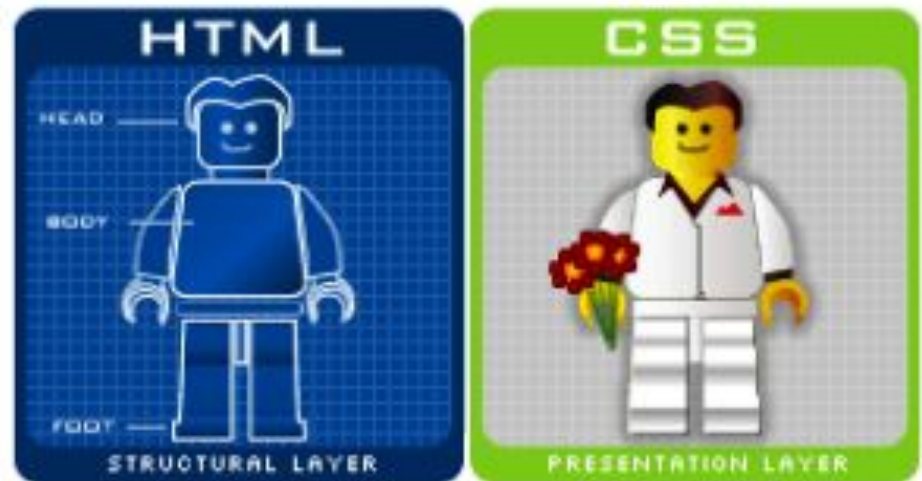
O telefone para contato é : **(41) 3721-8300**



Quem Somos

Responsabilidades

- **HTML** serve para estruturação do conteúdo
 - Somente isso. No HTML, esquecer a beleza.
 - Lembra?
- Tudo que é visual é responsabilidade do **CSS**
 - Cores
 - Tamanhos
 - Posicionamentos
 - Animações
 - ...



<http://portalwebdesigner.com/programacao/css>

Estilo Padrão

- Mas algumas partes da página ficaram estilizadas somente com HTML
 - Títulos, ênfases, links, listas
 - Por quê?
- O browser aplica estilos padrão para as tags
 - Você não vê, mas já é **CSS**
- Para deixarmos nosso site bonito:
 - Precisamos sobrescrever estes estilos

Antigamente

- Nos primórdios da web:
 - Estilizávamos a página direto no HTML
- Exemplo
 - Um título principal vermelho
 - `<h1>IFPR</h1>`
- Teste no navegador
 - Aproveite o momento
 - Este é a única vez na vida que você está permitido a utilizar a tag ****
 - Mas por quê?



Mas por quê?

- Tag `` também aceita outros atributos
 - face ← Estilo da fonte
 - size ← Tamanho
- E se colocarmos muitas `` no HTML?
 - Imagine um monte desses:
 - `Texto`
- Pensou na poluição do código final?
- Também estaríamos estilizando via HTML
 - Errado

A Dupla Dinâmica

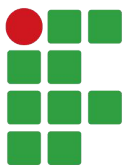
HTML



CSS



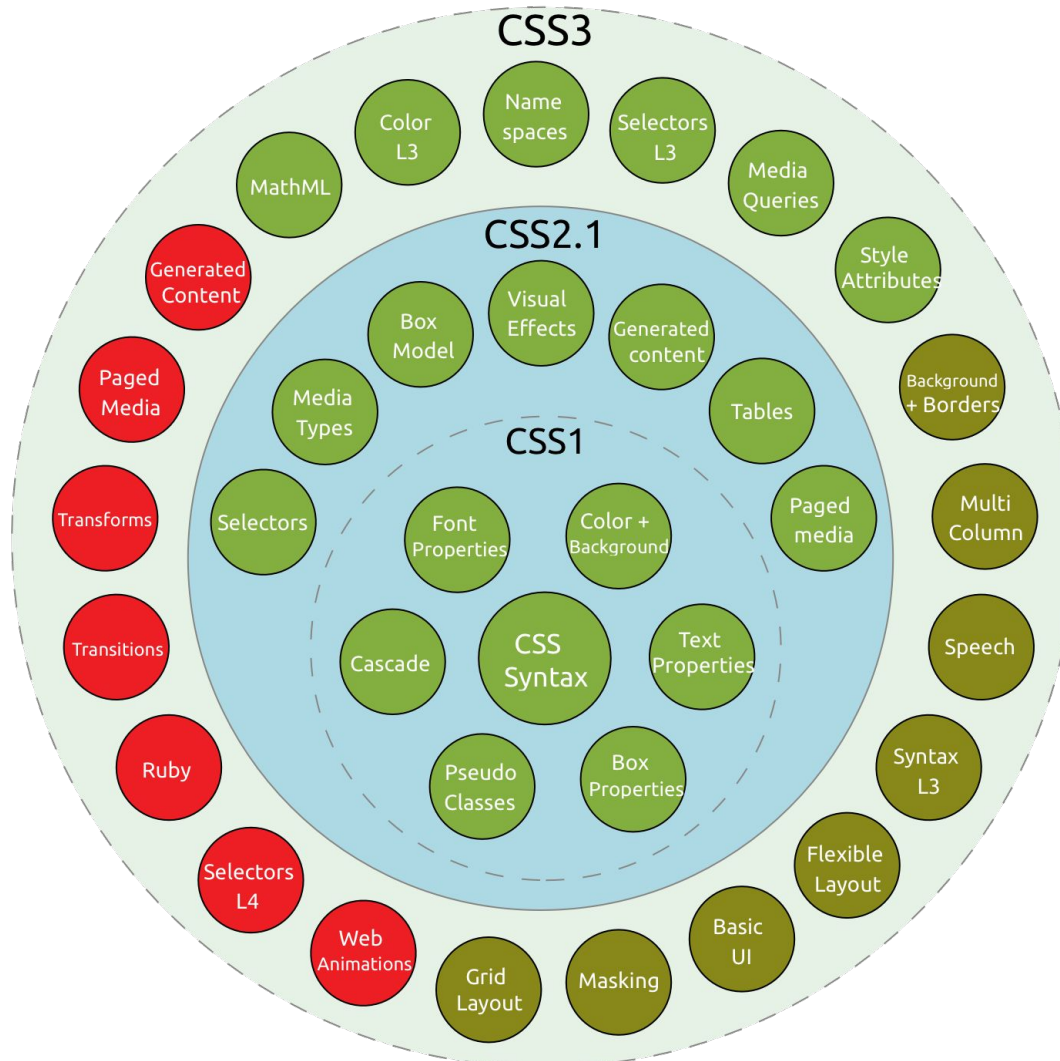
<http://portalwebdesigner.com/programacao/css>



CSS

- **Cascading Style Sheets**
 - Folha de Estilos em Cascata
- Lançada em 1996 (CSS1)
 - Versão atual → **CSS3**
 - História: <https://www.w3.org/Style/CSS20/history.html>
- Trabalha somente com estilização da página
 - Cores, posicionamento, espaçamentos, ...
 - Deixa o conteúdo para o HTML
- Define regras de estilização

Versões e Módulos



Definição Folhas de Estilos

- Existem **três** formas:

- Inline (dentro de um elemento específico)

```
<p style="color: red">Parágrafo</p>
```

- Tag style (dentro de head)

```
<style type="text/css">  
  p { color: blue; }  
</style>
```



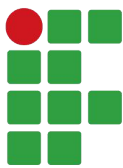
- Referenciar arquivo.css ← **Forma mais adequada**

```
<link rel="stylesheet" type="text/css"  
      href="estilo.css">
```

Vários Arquivos CSS

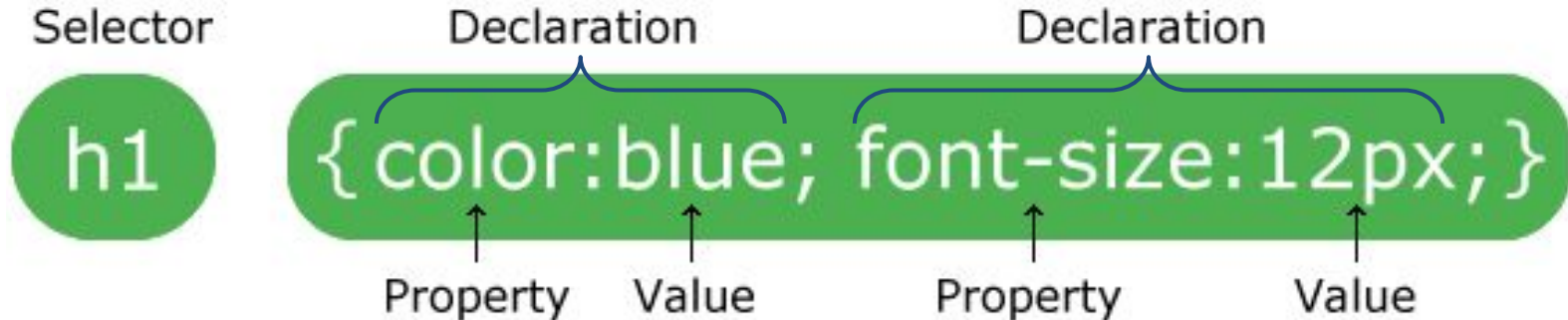
- Um documento HTML pode referenciar mais de um arquivo CSS
 - Serão importados na ordem especificada
- Exemplo

```
...  
<head>  
  <meta charset="utf-8">  
  <title>Page Title</title>  
  <link rel="stylesheet" type="text/css" href="principal.css">  
  <link rel="stylesheet" type="text/css" href="especial.css">  
  <link rel="stylesheet" type="text/css" href="outra_coisa.css">  
</head>  
...
```



Regra CSS

- Um arquivo .css contém diversas Regras CSS
 - Unidade mínima do CSS
- Uma Regra tem a sintaxe:



http://www.w3schools.com/css/css_syntax.asp



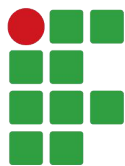
Regra CSS

- Seletor
 - Elementos nos quais a regra será aplicada
- Declaração
 - Um parâmetro de estilização
 - Composto por propriedade e valor
- Propriedade
 - Característica a ser estilizada
- Valor
 - Valor a ser aplicado à característica

Regras e Declarações

- Uma Regra CSS pode conter nenhuma, uma ou diversas **declarações**
 - Separadas por ponto e vírgula → ;

```
p {  
    color: blue;  
}  
  
h1 {  
    color: red;  
    font-size: 20px;  
}  
  
h2 {  
    color: yellow;  
    font-size: 110%;  
    background-color: red;  
    text-transform: uppercase;  
    font-family: Arial, Verdana;  
}
```

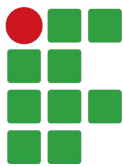


Seletor

- Usado para selecionar elemento(s)
 - Aplicamos as declarações para estilizar o(s) elemento(s) selecionados com ele
- Exemplos:
 - **Selecionar** todos os parágrafos, **mudar a cor do texto** para vermelho e **colocar em itálico**
 - **Selecionar** todos os links e **tirar o sublinhado**
 - **Selecionar** todos os itens de uma lista, **colocar uma margem** de 15 pixels, **apresentar todos na mesma linha** e **deixar o texto vermelho** quando o cursor do mouse estiver sobre eles

Colorindo Nosso Site

- Crie um arquivo **estilos.css**
- Vincule ele ao HTML do site do IFPR
- Estilizar
 - Títulos primários
 - Cor verde, fonte Arial (ou Verdana), 30 pixels de altura e em letras maiúsculas
 - Títulos secundários
 - Cor verde, fonte Arial (ou Verdana), 25 pixels de altura
 - Parágrafos
 - Alinhamento justificado, fonte Arial (ou Verdana)



Nosso CSS

- Provavelmente você terminou com algo parecido com isto em estilo.css:

```
h1 {  
    color: green;  
    font-family: Arial, Verdana;  
    font-size: 30px;  
    text-transform: uppercase;  
}  
h2 {  
    color: green;  
    font-family: Arial, Verdana;  
    font-size: 25px;  
}  
p {  
    font-family: Arial, Verdana;  
    text-align: justify;  
}
```

Otimizando

- Há declarações repetitivas
 - Por exemplo:
 - Cor e tipo da fonte em h1 e h2
- Podemos agrupá-las em somente uma regra

```
h1, h2 {  
    color: green;  
    font-family: Arial, Verdana;  
}  
h1 {  
    font-size: 30px;  
    text-transform: uppercase;  
}  
h2 {  
    font-size: 25px;  
}
```

Problema na Fonte?

- Repare em todo o texto da página
- Está com tipos de fonte diferentes?
 - Feio, né?
- Qual a solução?
 - Colocar font-family em todos elementos?
 - Não parece bom!
- Mover font-family para **seletor** body
 - Acontece a herança para os subelementos

```
body {  
    font-family: Arial, Verdana;  
}
```

Ainda Sobre Fontes...

- font-family define o tipo da fonte
 - Pode ser que uma fonte não exista no computador do nosso usuário
 - Podemos oferecer alternativas

- Exemplos:

- font-family: Arial, Verdana, "Open Sans", sans-serif;
- font-family: Cambria, serif;

Fonte SEM serifa
padrão do sistema

Fonte COM serifa
padrão do sistema

AaBbCc Tipo sem serifa
AaBbCc Tipo serifado
AaBbCc Serifas em vermelho

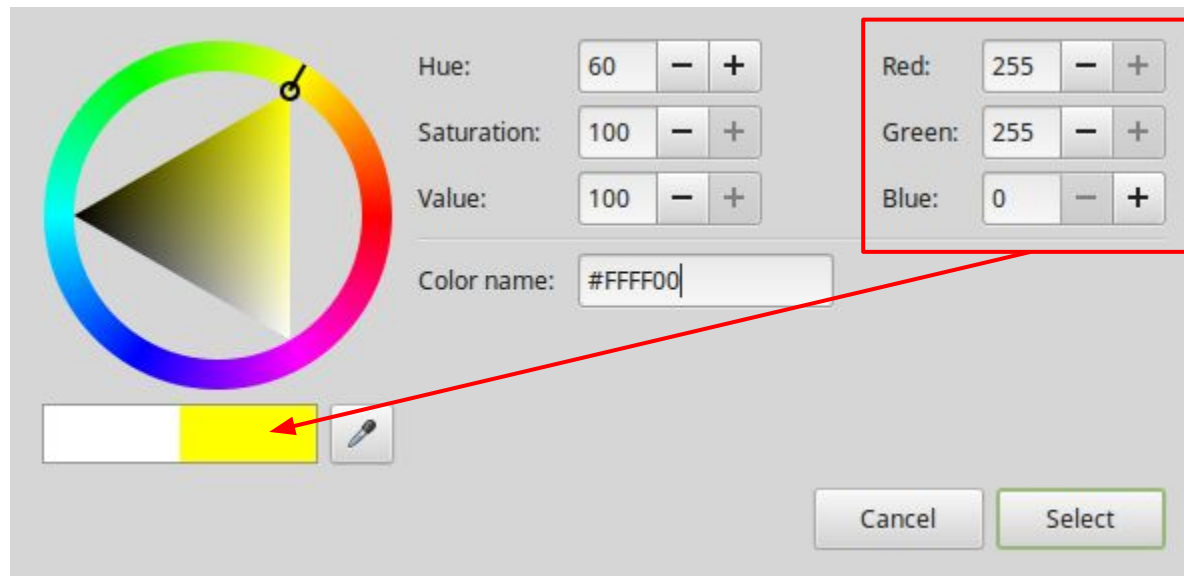


Cores

- Até agora usamos cores por constantes
 - red, green, yellow, blue, ...
- Estes nomes são apenas apelidos para cores
- Forma melhor definir cores:
 - Formato RGB (Red Green Blue)
 - Definimos a intensidade de cada uma das 3 cores
 - A junção delas nos dá a cor final

Cores RGB

- No RGB, cada cor (vermelho, verde e azul) é representada por uma escala de 0 a 255
 - 0 ← Ausência total da cor
 - 255 ← Presença total da cor



Cores no CSS

- Podemos representar cores de duas formas
 - `rgb(vermelho, verde, azul)`
 - `rgb(255, 255, 0)` ← Amarelo
 - `rgb(150, 50, 150)` ← Tom de Roxo
 - `rgb(171, 255, 191)` ← Verde Claro
 - Formato hexadecimal → `#RRGGBB`
 - `#ffff00` ← Amarelo
 - `#963296` ← Tom de Roxo
 - `#abffbf` ← Verde Claro

Cores: Exemplos de Uso

```
p {  
  color: #a02222;  
  background-color: #eeea40;  
}
```

```
h1 {  
  border-style: solid;  
  border-color: #db26dc;  
  color: #00ffbf;  
}
```

Prática

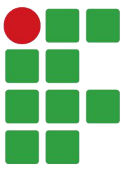
- Colorize nosso site do IFPR usando apenas o formato RGB Hexadecimal
 - Cor da fonte de elementos
 - Cor do fundo da página
 - Cor do fundo de elementos específicos
- Defina uma fonte principal e suas alternativas

Selecionando Elementos

- A forma como usamos o seletor nos obriga a estilizar **TODOS** os elementos com aquela tag
 - Por exemplo → Todos os <p>
 - Isto é um Seletor de Elemento (pela tag)
- Nem sempre queremos isso
 - Algumas vezes queremos apenas **aplicar uma regra** a um elemento específico ou conjunto de elementos específicos

Selecionando Elementos

- Existe solução para isto
- Voltando ao HTML, podemos nos valer de dois atributos de tags
 - **id** e **class**
- Um **ID** (identificador) serve para definir um elemento que é ÚNICO na página
- Uma **Classe** serve para marcar diversos elementos que compartilham características



ID

- Nós já utilizamos no nosso exemplo
 - `<h3 id="instituto">0 Instituto</h3>`
 - Aqui o objetivo era fazer links internos
- Este **ID** NÃO PODE existir em outro elemento
- No CSS, utilizamos **cerquilha** para selecionar

Chamamos de Seletor de ID

```
#instituto {  
    color: #cb0008;  
}
```

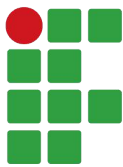


Classe

- Podemos utilizar em diversos elementos
 - `<ul class="cursos">...`
 - `<ul class="cursos">...`
- A **Classe** PODE existir em outros elementos
- No CSS, utilizamos **ponto** para selecionar

Chamamos de Seletor de Classe

```
.cursos {  
    color: #0004a3;  
}
```



Seletor *

- O seletor asterisco (*) permite estilizar TODOS os elementos do HTML de uma vez
- Muito utilizado para “zerar” margens
 - Colocar no começo do documento CSS:

```
* {  
    margin: 0;  
    padding: 0;  
}
```

Elimina qualquer margem e preenchimento de TODOS os elementos existentes

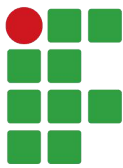
Nas linhas abaixo será necessário redefinir qualquer margem e preenchimento que quiser utilizar

Várias Classes

- Um elemento pode ter mais de uma classe
 - Significa que o elemento possui as características de ambas as classes
 - `<p class="importante principal">...</p>`
- Os seletores CSS **.importante** e **.principal** irão selecionar e estilizar o elemento `<p>`

Estilos em Cascata

- Como o nome do CSS diz, estamos trabalhando com estilos em cascata
 - Significa que os estilos são adicionados aos elementos HTML de cima para baixo de acordo com a aparição no documento CSS
 - Se duas diferentes declarações aplicarem a mesma propriedade sobre um elemento, **VENCE a última (sobrescreve)**



Exemplos

```
h1 {  
background-color: #ff0000;  
font-family: Cambria, serif;  
}
```

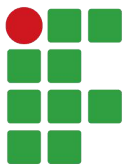
```
h1 {  
background-color: #0000ff;  
}
```

Esta declaração VENCE, pois
aparece por último

Esta declaração segue válida,
pois não foi sobrescrita
posteriormente

Especificidade

- Além da ordem no documento, temos outra forma **(mais importante)** de cascadeamento
 - Seletores mais específicos **VENCEM** os seletores mais genéricos (para uma propriedade repetida)
- Especificidade
 - Seletor * ← Mais GENÉRICO
 - Seletor de elemento
 - Seletor de classe
 - Seletor de ID ← Mais ESPECÍFICO



Exemplos

```
<h1 class="principal">Título Principal</h1>
```

```
.principal {  
  background-color: #ff0000;  
}
```

Esta declaração VENCE,
pois .principal é mais
específico que h1

```
h1 {  
background-color: #0000ff;  
font-family: Cambria, serif;  
}
```

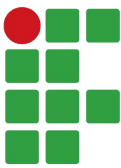
Esta declaração segue válida, pois não foi sobrescrita
posteriormente ou por uma regra mais específica

Box Model

- Todo elemento HTML é visto como uma caixa
- O conteúdo é envolto por três camadas
 - Preenchimento: **Padding**
 - Borda: **Border**
 - Margem: **Margin**



https://www.w3schools.com/css/css_boxmodel.asp



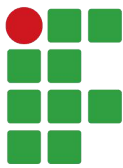
Box Model

- **Content** (Conteúdo)
 - Conteúdo útil do elemento
- **Padding** (preenchimento)
 - Área transparente entre o conteúdo e a borda
- **Border** (borda)
 - Linha ao redor do preenchimento e conteúdo
- **Margin** (margem)
 - Área transparente ao redor da borda
 - Separa o elemento dos demais elementos

Padding, Border e Margin

- No CSS, podemos estilizar todos eles
- Tamanho:

```
p {  
    padding: 5px;  
    border: 10px;  
    margin: 15px;  
}
```



Border

- Para border funcionar, precisamos de mais informações
 - Cor
 - Estilo: sólida, pontilhada, tracejada, ...
 - [Ver todos](#)
- Exemplo:

```
p {  
    border-width: 10px;  
    border-style: solid;  
    border-color: #e8ec73;  
}
```


Arestas Específicas

- As três propriedades aceitam a manipulação de uma aresta específica
 - top, right, bottom, left
- Exemplos:

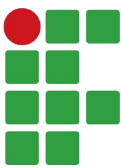
```
p {  
    border-top: 1px;  
    border-top-style: dashed;  
    margin-bottom: 10px;  
    padding-left: 5px;  
    padding-right: 2px;  
}
```

Border Shorthand

- Para abreviar, podemos definir as três propriedades de border ao mesmo tempo
- Exemplos:

```
p {  
    border: 2px solid #ffffff;  
}
```

```
h1 {  
    border-bottom: 1px dashed #f58787;  
}
```



Tamanhos

- Podemos manipular o tamanho de elementos
 - **Largura** (width)
 - `width: 10px;` ← Fixa em 10 pixels
 - `width: 50%;` ← 50% da largura do elemento-pai
 - **Altura** (height)
 - `height: 50px;` ← Fixa em 50 pixels
- Obs: Nem sempre devemos manipular a altura dos elementos

```
img {  
    width: 150px;  
    height: auto;  
}
```