



# Framework Yii

**Instrutor: Luciano Moraes Da Luz Brum.**

**Email: [lucianobrum18@gmail.com](mailto:lucianobrum18@gmail.com)**

**Sala: 3143.**

# Programação do Minicurso

- **Módulo 1:** Conceitos Básicos de Desenvolvimento Web. Princípios Básicos sobre **HTML 5.0**. Princípios Básicos de **CSS**. Princípios Básicos das Linguagens de Programação **PHP** e **JS**. Criando aplicações simples.
- **Módulo 2:** Conceitos básicos sobre banco de dados. Sistemas gerenciadores de banco de dados. Como acessar banco de dados usando **PHP**. Criando aplicações básicas.
- **Módulo 3:** O uso de frameworks no desenvolvimento de aplicações web. Vantagens e desvantagens. Conceitos básicos sobre o **framework Yii (2.0 ou 1.1.16)**. Exemplos básicos com **Yii**.
- **Módulo 4:** Mais exemplos básicos. Desenvolvimento com **Yii** e integração dos conceitos.
- **Módulo 5:** Criando uma aplicação com o framework **Yii** e integração dos conceitos.

**Módulo I:**

**Conceitos Básicos de Aplicações Web**

# Roteiro

- Conceitos Básicos sobre desenvolvimento web.
- Princípios Básicos da Linguagem HTML 5.0.
- Princípios Básicos de CSS.
- Princípios Básicos da Linguagem de Programação PHP e JS.

# **Conceitos Básicos sobre Desenvolvimento Web**



# Conceitos Básicos

- **Desenvolvimento web** é o termo utilizado para descrever o desenvolvimento de sites, na Internet ou numa Intranet;
- **Web Designer:** Desenvolvedor do Layout;
- **Web Developer:** Desenvolvedor de sistemas;



# BRAIN HEMISPHERE AT WORK

■ WEB DESIGNER  
■ WEB DEVELOPER

RIGHT



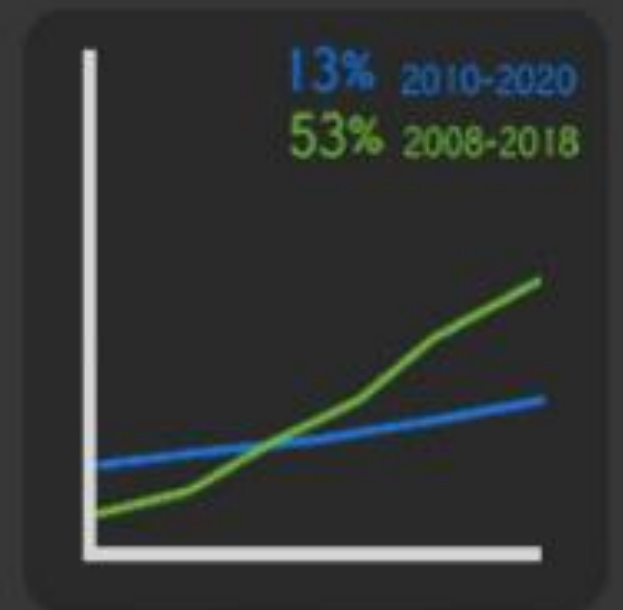
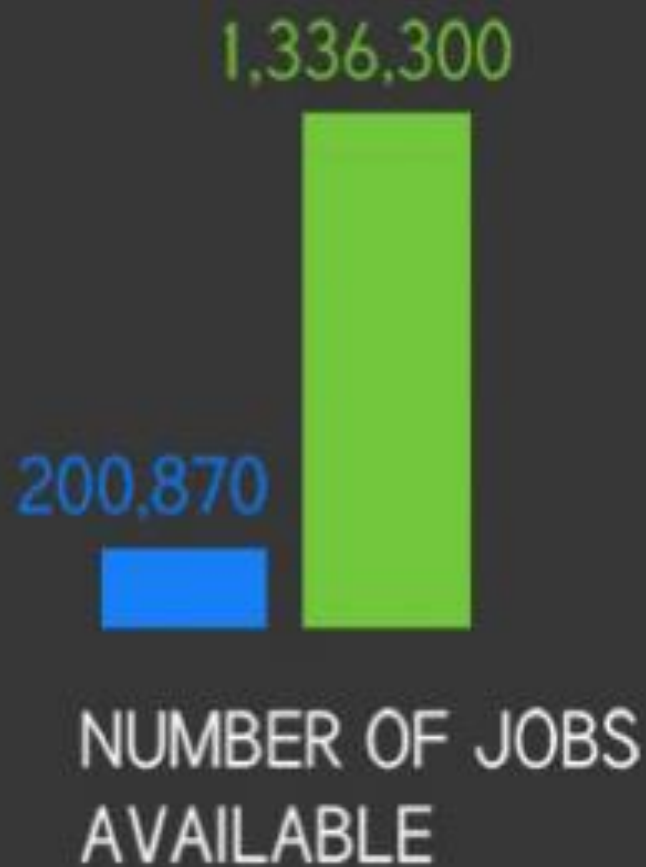
INTUITION  
CREATIVITY  
IMAGINATION

LEFT



LOGIC  
LINEAR THINKING  
TECHNICAL

# JOB STATISTICS





# EDUCATIONAL REQUIREMENT



UNDERGRAD/ GRADUATE  
GRAPHIC DESIGN  
FINE ARTS  
SIMILAR FIELD  
PORTFOLIO



An undergrad or graduate degree in graphic design, fine arts, multimedia production, or a similar field of study in the creative arts is required education for most graphic design positions. Comprehensive portfolio of your best design projects.

PROGRAMMING EXPERIENCE

BEST WEBPAGES  
APPLICATIONS  
PORTFOLIO



Hands-on programming experience is the key requirement for securing work as a web developer. Hiring managers for web development positions will typically ask to see a portfolio with samples of the best web pages and applications you've created.



# Café Caçara

HOME

CAFÉ CAÇARA

PRODUTOS

MAQUINAS

HISTÓRIA DO CAFÉ

RECEITAS

CAMPANHA ABIC

CONTATO

BUSCA... 



QUALIDADE CERTIFICADA

## Café Caçara

O café também é usado como ingrediente de muitas delícias da culinária. Descubra novas possibilidades de aproveitar ao máximo todo esse sabor.

Saiba +

## Receitas

O café também é usado como ingrediente de muitas delícias da culinária. Descubra novas possibilidades de aproveitar ao máximo todo esse sabor.

Saiba +

## História do Café

O café também é usado como ingrediente de muitas delícias da culinária. Descubra novas possibilidades de aproveitar ao máximo todo esse sabor.

Saiba +

FEITO

PARA

VOÇÊ



## Overview

Dashboard

Cascade

Pages

Utility

## Components

UI Elements

Tables

Unlimited Menu

Forms

Charts

Maps

Icons

## Dashboard

Search through site



Sales

This Week

233

43%



Downloads

This Year

233

52%

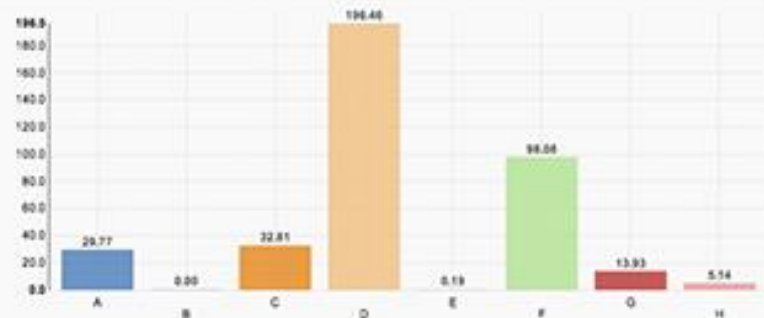


Comments

This Month

2,987

22%



Active Users

1500ms

33

23%



Total Users

This Year

5033

23%



Profit

Projected

\$98,233

48%

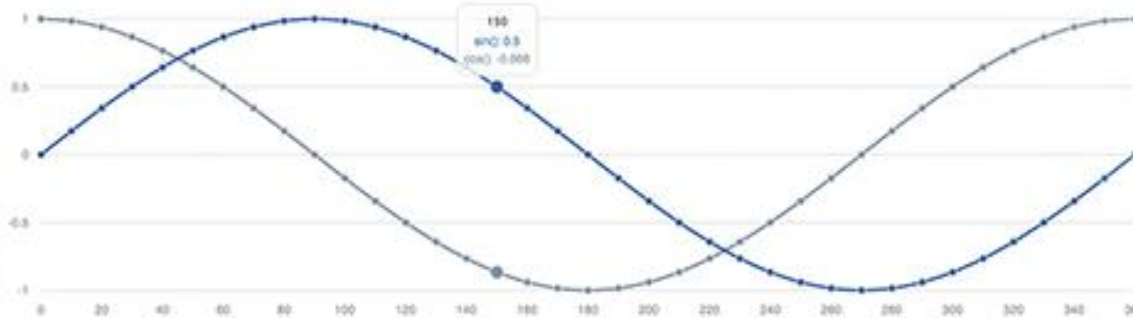


Returns

This Year

33

23%



## Users

User ID	Email	Status
Vijay	me@example.com	Approved
Anusha	you@example.com	Pending

## Analytics

↑ 68%





# Conceitos Básicos

- O desenvolvimento refere-se a um processo de construção e testes do software específico para a web, com a finalidade de se obter um conjunto de programas, que satisfazem as funções pretendidas, quer em termos de usabilidade dos usuários ou compatibilidade com outros programas existentes.
- O desenvolvimento web pode variar desde **simples páginas estáticas** a **aplicações ricas**, comércios eletrônicos ou redes sociais.

# Conceitos Básicos

- **Vantagens no uso de aplicações web:**
  - Os custos são menores do que sistemas desktop, dependendo da situação;
  - Possibilidade de integrar sistemas da empresa na web;
  - Empresa não precisa investir em infra estrutura. Existem provedores que por custos mensais interessantes, disponibilizam com qualidade e segurança toda estrutura necessária para hospedagem de sistemas web;
  - O acesso a sistemas web pode ser feito de qualquer computador ou dispositivo móvel conectado a web com segurança, pois seu desenvolvimento já prevê acessos restritos;

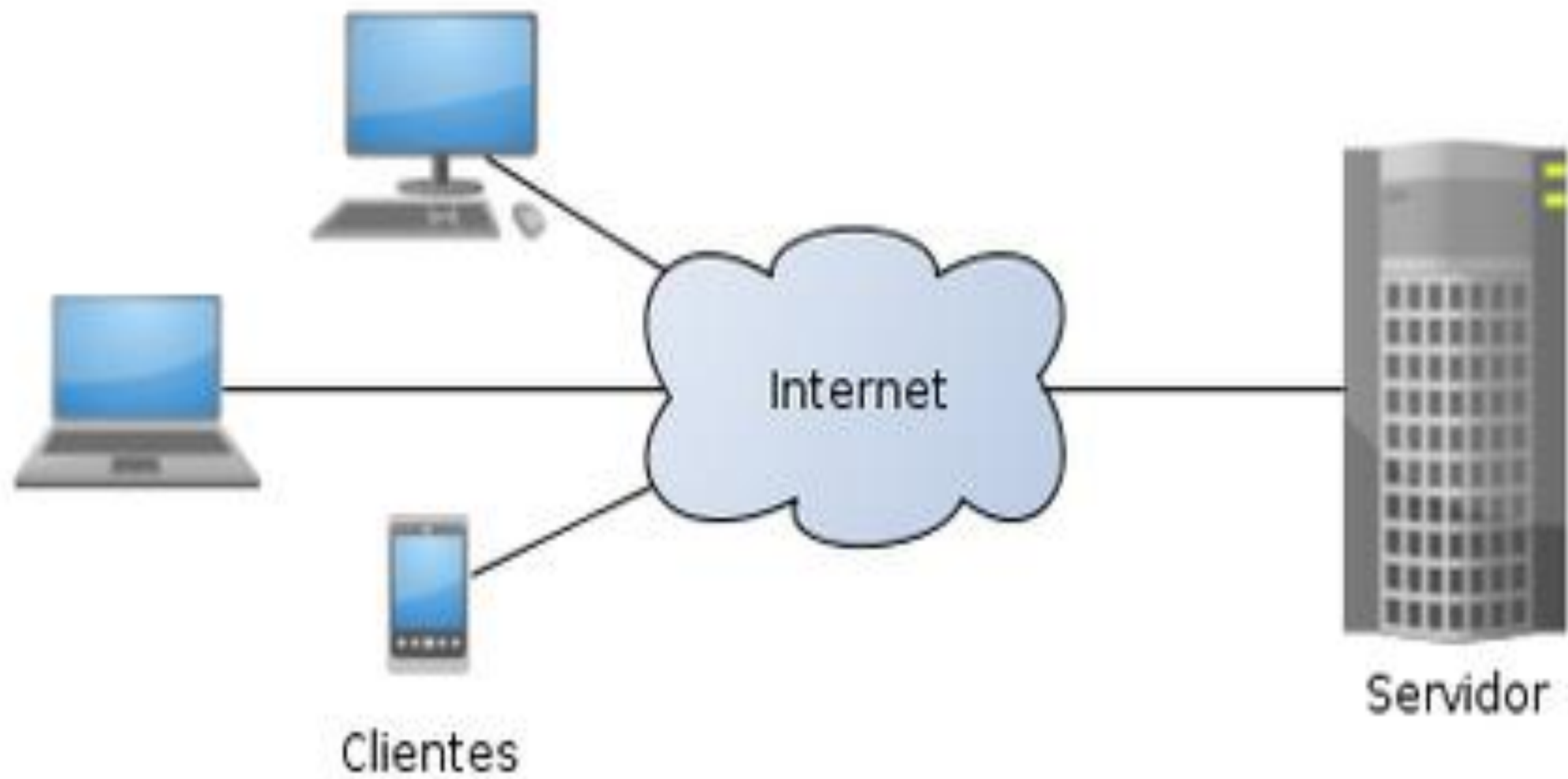
# Conceitos Básicos

- **Vantagens no uso de aplicações web:**
- Não precisa ser desenvolvido todo de uma só vez. É possível fazer o desenvolvimento por partes, o que fará com que o sistema acompanhe a evolução da sua empresa.
- Segurança. Backups periódicos são realizados automaticamente. A integridade dos dados e segurança das informações é garantida por acessos restritos de usuários autorizados com perfis previamente definidos do que ele pode acessar ou não.
- Flexibilidade.



# Conceitos Básicos

- **Cliente: Quem requisita um recurso ou serviço. Características:**
  - Inicia pedidos para servidores;
  - Espera por respostas e recebe respostas;
  - Conecta-se a um pequeno número de servidores de uma só vez;
  - Normalmente interage diretamente com os usuários finais através de qualquer interface com o usuário;
  - Utiliza recursos da rede para comunicação (barramentos, switches, hubs, roteadores, etc...);
- **Servidor: Fornecedores de um recurso ou serviço. Características:**
  - Sempre espera por um pedido de um cliente;
  - Atende os pedidos e, em seguida, responde aos clientes com os dados solicitados;
  - Pode se comunicar com outros servidores para atender uma solicitação específica do cliente;
  - Fornece recursos de rede;
  - Estrutura o sistema;



# Conceitos Básicos

- Boa parte das aplicações web precisam armazenar uma grande quantidade de informações (Ex: login e senha de usuários de uma rede social, informações de acesso ao portal de uma universidade, informações pessoais, etc).
- Todas estas informações são armazenadas em bancos de dados **nos servidores**.

# Conceitos Básicos

- O acesso a estes dados se dá por meio da utilização de um SGBD (Sistema Gerenciador de Banco de Dados). Ex: SGBD Oracle, SGBD MySQL, SGBD SQL Server, etc.
- Cada SGBD implementa um banco de dados (ou vários) de uma maneira diferente, mas para o usuário isso é transparente, pois a linguagem de acesso aos dados é a mesma, o SQL.



# Princípios Básicos da Linguagem HTML 5.0



# HTML 5.0

- A única linguagem que o navegador consegue interpretar para a exibição de conteúdo é o HTML (*HyperText Markup Language*).
- Para iniciar a exploração do HTML, vamos imaginar o seguinte caso:
- O navegador realizou uma requisição e recebeu como corpo da resposta o seguinte conteúdo:

# HTML 5.0

## Mirror Fashion.

Bem-vindo à Mirror Fashion, sua loja de roupas e acessórios.

- Confira nossas promoções.
- Receba informações sobre nossos lançamentos por email.
- Navegue por todos nossos produtos em catálogo.
- Compre sem sair de casa.

## Mirror Fashion.

Bem-vindo à Mirror Fashion, sua loja de roupas e acessórios.

- Confira nossas promoções.
- Receba informações sobre nossos lançamentos por email.
- Navegue por todos nossos produtos em catálogo.
- Compre sem sair de casa.

# HTML 5.0

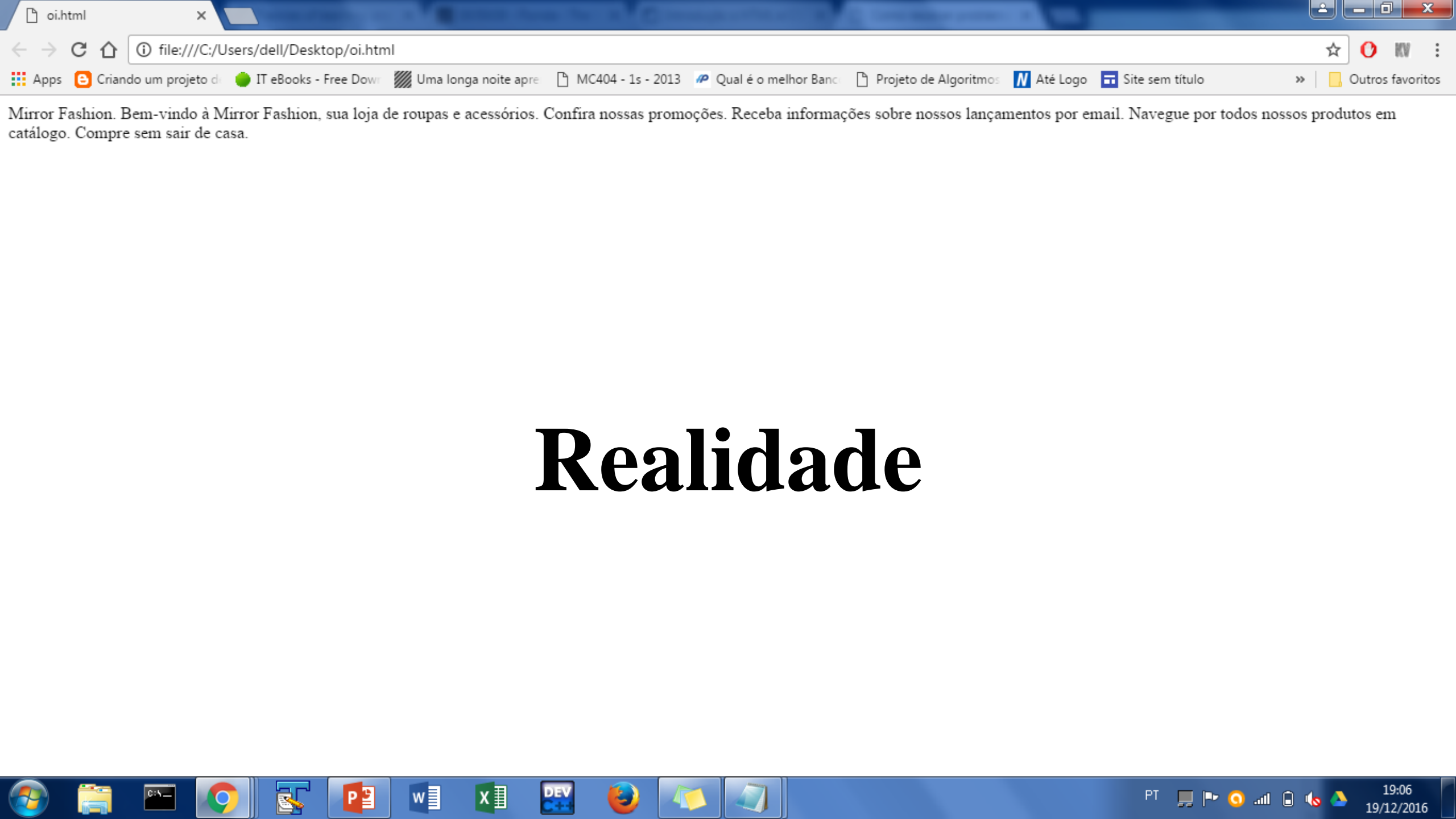
- Copiem este texto em um bloco de notas e salvem o arquivo com o nome **index.html**.
- Abram o arquivo com o navegador de preferência (IE, Chrome, Firefox, Opera, Edge, etc).
- Qual foi o resultado?

# Mirror Fashion.

Bem-vindo à Mirror Fashion, sua loja de roupas e acessórios.

- Confira nossas promoções.
- Receba informações sobre nossos lançamentos por email.
- Navegue por todos nossos produtos em catálogo.
- Compre sem sair de casa.

# Expectativa



# Realidade

# HTML 5.0

- Parece que obtemos um resultado um pouco diferente do esperado, não?
- Algumas regras devem ser seguidas caso desejemos que esse texto seja exibido com alguma formatação, para facilitar a leitura pelo usuário final.



# HTML 5.0

- Copiem este texto em um bloco de notas e salvem o arquivo com o nome **index2.html**. Ao salvar, marquem a codificação utf-8. Abram o arquivo com o navegador de preferência.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Mirror Fashion</title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1>Mirror Fashion.</h1>
    <h2>Bem-vindo à Mirror Fashion, sua loja de roupas e acessórios.</h2>
    <ul>
      <li>Confira nossas promoções.</li>
      <li>Receba informações sobre nossos lançamentos por email.</li>
      <li>Navegue por todos nossos produtos em catálogo.</li>
      <li>Compre sem sair de casa.</li>
    </ul>
  </body>
</html>
```

# Mirror Fashion.

Bem-vindo à Mirror Fashion, sua loja de roupas e acessórios.

- Confira nossas promoções.
- Receba informações sobre nossos lançamentos por email.
- Navegue por todos nossos produtos em catálogo.
- Compre sem sair de casa.

# Expectativa

# Mirror Fashion.

Bem-vindo à Mirror Fashion, sua loja de roupas e acessórios.

- Confira nossas promoções.
- Receba informações sobre nossos lançamentos por email.
- Navegue por todos nossos produtos em catálogo.
- Compre sem sair de casa.

# Realidade

# HTML 5.0

- Agora, o resultado é exibido de maneira agradável e legível.
- Para isso, tivemos que utilizar algumas marcações do HTML. Essas marcações são chamadas de **tags**, e elas basicamente dão **significado** ao texto contido entre sua abertura e fechamento.
- Mesmo assim, as informações não apresentam atrativo estético e, nessa deficiência do HTML, reside o maior desafio do programador *front-end*.

# Syntaxe des TAGs



# Sintaxe das Tags

- O HTML é um conjunto de **tags** responsáveis pela marcação do conteúdo de uma página no navegador.
- No código de antes, vimos o uso da tag `<h1>`:



`<h1> Mirror Fashion </h1>`

- Uma tag é definida com os caracteres `<` e `>` e seu nome. Tags que possuem conteúdo precisam de tag de fechamento para informar onde o conteúdo acaba.



# Sintaxe das Tags

- Algumas tags podem receber atributos dentro de sua definição.
- Por exemplo, para definir uma imagem, usamos a tag `<img>`, e para indicar qual imagem carregar, usamos o atributo `src`:



```
<img src = ../imagens/casa_da_praia.jpg>
```

- Obs: a tag `img` não possui conteúdo. Portanto, não é necessária uma tag de fechamento, como antes no `h1`.

# Sintaxe das Tags

- A tag **<figure>** define uma imagem com a conhecida tag **<img>**. Além disso, permite adicionar uma legenda para a imagem por meio da tag **<figcaption>**.

```
<figure>  
    
  <figcaption>Fuzz Cardigan por R$ 129,90</figcaption>  
</figure>
```

- O atributo “*alt*” é um texto alternativo para a imagem caso não possa ser carregada .

# Sintaxe das Tags

- `<!DOCTYPE html>`
- DOCTYPE não é uma tag HTML, mas uma instrução especial. Ela indica para o navegador qual **versão do HTML** deve ser utilizada para renderizar a página.
- Utilizaremos `<!DOCTYPE html>`, que indica para o navegador a utilização da versão mais recente do HTML - a versão 5, atualmente.

# Sintaxe das Tags

- `<html>`
- Dentro desta tag, devemos declarar outras 2 tags: `<head>` e `<body>`.
- Um documento HTML válido deve possuir as tags `<html>`, `<head>`, `<body>` e a instrução `<!DOCTYPE html>`.

# Sintaxe das Tags

- `<head>`
- Contém informações do documento que são apenas de interesse para o navegador. Nenhuma informação é exibida na área do navegador.
- `<title>`
- Permite especificar o título do documento que é exibido na barra de título do navegador. Deve aparecer dentro da tag `<head>`.

# Sintaxe das Tags

- Devemos definir a codificação de caracteres para a página, chamado de **charset** ou **encoding**.
- O UTF-8 é a recomendação atual para encoding na web por ser amplamente suportada em navegadores e editores e por ser compatível com praticamente todos idiomas do mundo. Usamos a tag <meta> para isso, que deve estar dentro da tag <head>, conforme a seguir:



```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Mirror Fashion</title>
```

```
    <meta charset="utf-8">
```

```
  </head>
```

# Sintaxe das Tags

- `<body>`: Contém o corpo do documento, que é exibido na página pelo navegador. É necessário que o `<body>` tenha, pelo menos, um elemento filho. Neste exemplo, usamos a tag `<h1>`, que indica que seu conteúdo é um título.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Mirror Fashion</title>
```

```
<meta charset="utf-8">
```

```
</head>
```

```
<body>
```

```
<h1> Mirror Fashion </h1>
```

```
</body>
```

```
</html>
```



# Sintaxe das Tags

- `<h1>`: Indica que seu conteúdo é um título. As tags podem ser de `<h1>` até `<h6>`, seguindo a ordem de importância, sendo que `<h1>` é o título mais importante.
- **Para exibir qualquer texto no navegador, é necessário que este conteúdo esteja em uma tag filha de `<body>`.**
- `<p>`: Marcação mais indicada para textos comuns é esta tag de parágrafo. Se você tiver vários parágrafos, use várias tags `<p>` para separa-los:

`<p>` My name is Oliver Queen. `</p>`

`<p>` After five years in hell... `</p>`

# Sintaxe das Tags

- `<b>`: Tag que indica que seu conteúdo será em negrito.
- `<i>`: Tag que indica que seu conteúdo será em itálico.
- `<ul>`: Tag que cria listas não-ordenadas. É usada em conjunto com a tag `<li>` que indica cada item da lista.

`<body>`

`<h1>Mirror Fashion.</h1>`

`<h2>Bem-vindo à Mirror Fashion, sua loja de roupas e acessórios.</h2>`

`<ul>`

`<li>Confira nossas promoções.</li>`

`<li>Receba informações sobre nossos lançamentos por email.</li>`

`<li>Navegue por todos nossos produtos em catálogo.</li>`

`<li>Compre sem sair de casa.</li>`

`</ul>`

`</body>`

# Exercício: Primeira Página

- A primeira página que desenvolveremos para a Mirror Fashion será a Sobre, que explica detalhes sobre a empresa, apresenta fotos e a história.
- Recebemos o design já pronto, assim como os textos. Nosso trabalho, como desenvolvedores de front-end, é codificar o HTML e CSS necessários para esse resultado.

## A Mirror Fashion

A Mirror Fashion é a maior empresa comércio eletrônico no segmento de moda em todo o mundo. Fundada em 1932, possui filiais em 124 países, sendo líder de mercado com mais de 90% de participação em 118 deles.

Nosso centro de distribuição fica em [Jacarezinho, no Paraná](#). De lá, saem 48 aviões que distribuem nossos produtos às casas do mundo todo. Nosso centro de distribuição:



Centro de distribuição da Mirror Fashion

Compre suas roupas e acessórios na Mirror Fashion. Acesse [nossa loja](#) ou entre em [contato](#) se tiver dúvidas. Conheça também nossa [história](#) e nossos [diferenciais](#).

### História

A fundação em 1932 ocorreu no momento da descoberta econômica do interior do Paraná. A família Pelho, tradicional da região, investiu todas as suas economias nessa nova iniciativa, revolucionária para a época. O fundador *Eduardo Simões Pelho*, dotado de particular visão administrativa, guiou os negócios da empresa durante mais de 50 anos, muitos deles ao lado de seu filho *E. S. Pelho Filho*, atual CEO. O nome da empresa é inspirado no nome da família.

O crescimento da empresa foi praticamente instantâneo. Nos primeiros 5 anos, já atendia 18 países. Bateu a marca de 100 países em apenas 15 anos de existência. Até hoje, já atendeu 740 milhões de usuários diferentes, em bilhões de diferentes pedidos.

O crescimento em número de funcionários é também assombroso. Hoje, é a maior empregadora do Brasil, mas mesmo após apenas 5 anos de sua existência, já possuía 30 mil funcionários. Fora do Brasil, há 240 mil funcionários, além dos 890 mil brasileiros nas instalações de Jacarezinho e nos escritórios em todo país.

Dada a importância econômica da empresa para o Brasil, a família Pelho já recebeu diversos prêmios, homenagens e condecorações. Todos os presidentes do Brasil já visitaram as instalações da Mirror Fashion, além de presidentes da União Européia, Ásia e o secretário-geral da ONU.



Família Pelho

### Diferenciais

- Menor preço do varejo, garantido
- Se você achar uma loja mais barata, leva o produto de graça
- Pague em reais, dólares, euros ou bitcoins
- Todas as compras com frete grátis para o mundo todo
- Maior comércio eletrônico de moda do mundo
- Atendimento via telefone, email, chat, twitter, facebook, carta, fax e telegrama
- Presente em 124 países
- Mais de um milhão de funcionários em todo o mundo

## Compre já em nossa loja!

# Exercício: Primeira Página

- Link: <https://s3.amazonaws.com/caelum.com.br/caelum-arquivos-curso-web.zip>
- Copie a pasta mirror-fashion de dentro da pasta Caelum/43 para a área de trabalho de sua máquina.
- Verifique a pasta img, agora cheia de arquivos do design do site. Além desta pasta, teremos as pastas js e css, que ainda não usaremos.



# Exercício: Primeira Página

- Dentro da pasta mirror-fashion, vamos criar o arquivo sobre.html com a estrutura básica contendo o DOCTYPE e as tags html, head e body....
- Feito isso, observem o resultado final. Na sequência, veremos como estilizar nossas páginas com CSS e nos aproximarmos mais do resultado final esperado.

# Estilizando com CSS



# CSS

- O navegador possui um estilo padrão para cada uma das tags que usamos.
- Para fazer sites dinâmicos e mais “bonitos”, precisamos customizar o design dos elementos da página.
- Antigamente isso era feito no próprio HTML. Hoje em dia não é mais usado e é considerada uma “má prática”.

# CSS

- Então surge o **CSS** (Cascading Style Sheets) como uma linguagem com objetivo único de cuidar da estilização da página.
- O CSS é bem mais robusto para estilização que o HTML e agora, **regras de formatação visual não são misturadas com conteúdo de texto no HTML.**
- Enfim, como utilizar o CSS no meu site?

# CSS

- A sintaxe do CSS tem estrutura simples: é uma declaração de propriedades e valores separados por um sinal de dois pontos ":", e cada propriedade é separada por um sinal de ponto e vírgula ";" da seguinte maneira:



```
background-color: yellow;  
color: blue;
```

- Essas propriedades podem ser declaradas de **três maneiras diferentes**.



# CSS

## 1. Atributo style (não mais usado, por misturar com HTML):

```
<p style="color: blue; background-color: yellow;">
```

My name is Barry Allen and I am the slowest man alive !!

```
</p>
```

# CSS

**2. Tag style:** A outra maneira de se utilizar o CSS é declarando suas propriedades dentro de uma tag `<style>`.

- Como estamos declarando as propriedades visuais de um elemento em outro lugar do nosso documento, precisamos indicar de alguma maneira a **qual** elemento nos referimos.
- Fazemos isso utilizando um **seletor CSS**. É basicamente uma forma de buscar certos elementos dentro da página que receberão as regras visuais que queremos.

# CSS

- No exemplo a seguir, usaremos o seletor que pega todas as tags <p> e altera sua cor e *background*:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Sobre a Mirror Fashion</title>
    <style>
      p {
        background-color: yellow;
        color: blue;
      }
    </style>
  </head>
```

# CSS

<body>

<p>

O conteúdo desta tag será exibido em azul com fundo amarelo!

</p>

<p>

<strong>Também</strong> será exibido em azul com fundo amarelo!

</p>

</body>

</html>

# CSS

- O código anterior da tag `<style>` indica que estamos alterando a cor e o fundo de **todos** os elementos com tag `p`.
- Dizemos que selecionamos esses elementos pelo nome de sua tag, e aplicamos certas propriedades CSS apenas neles.

# CSS

3. Arquivo externo: A terceira maneira de declararmos os estilos é com um **arquivo externo com a extensão .css**.
- Para que seja possível declarar nosso CSS em um arquivo à parte, precisamos indicar em nosso documento HTML uma **ligação** entre ele e a folha de estilo.
  - Além da melhor organização do projeto, a folha de estilo externa traz ainda as vantagens de **manter nosso HTML mais limpo** e do **reaproveitamento** de uma mesma folha de estilos para diversos documentos.



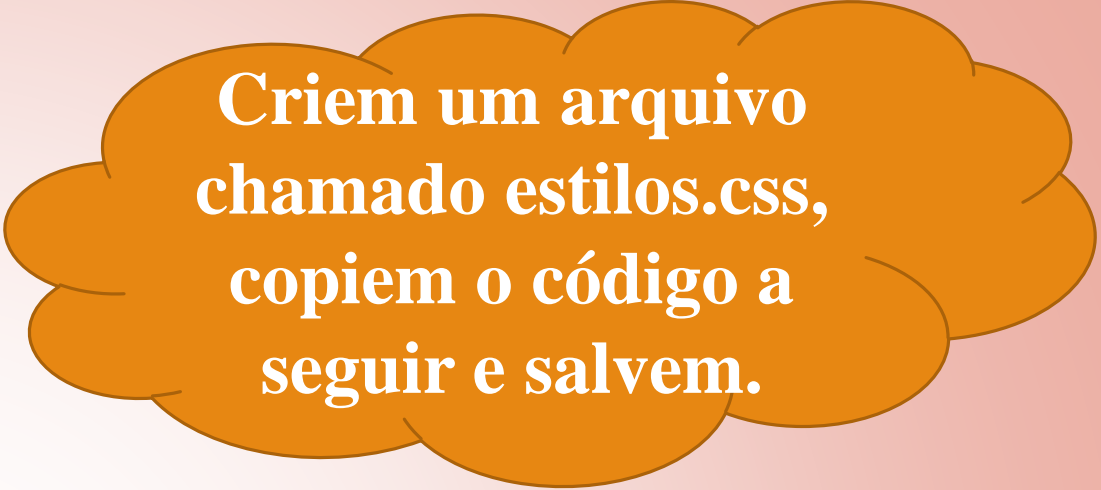
# CSS

- A **indicação de uso** de uma folha de estilos externa deve ser feita dentro da tag `<head>` do nosso documento HTML:
- Reproduza o exemplo a seguir:

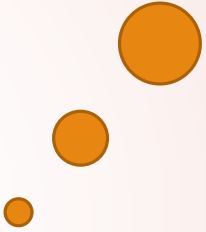
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Sobre a Mirror Fashion</title>
    <link rel="stylesheet" href="estilos.css">
  </head>
  <body>
    <p>
      O conteúdo desta tag será exibido em azul com fundo amarelo!
    </p>
    <p>
      <strong>Também</strong> será exibido em azul com fundo amarelo!
    </p>
  </body>
</html>
```



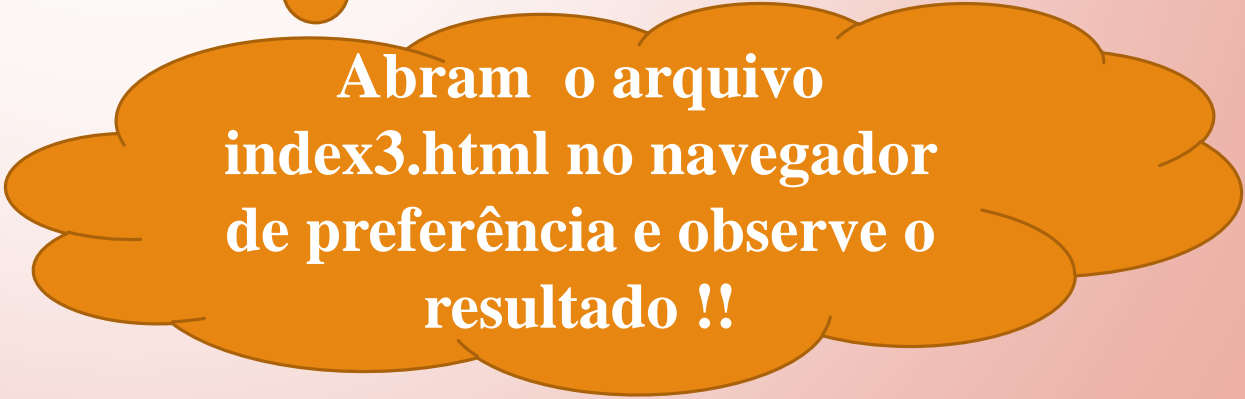
Salvem o arquivo  
com o nome  
index3.html



Criem um arquivo  
chamado estilos.css,  
copiem o código a  
seguir e salvem.



```
p {  
  color: blue;  
  background-color: yellow;  
}
```



Abram o arquivo  
index3.html no navegador  
de preferência e observe o  
resultado !!

# CSS

- Até o momento vimos as propriedades:
  - **color;**
  - **background-color;**
- Temos ainda a propriedade **font-family**: permite alterar as fontes do texto no navegador. Podemos alterar a fonte ou a família de fontes utilizada.

# CSS

```
h1 {  
  font-family: serif;  
}
```

Ex: Times, Times New Roman

```
h2 {  
  font-family: sans-serif;  
}
```

Ex: Arial, Helvética

```
p {  
  font-family: monospace;  
}
```

Ex: Courier

# CSS

Neste caso, a propriedade definida será aplicada em todo documento no navegador, pois alteramos a tag <body>

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

Primeiro o navegador buscará pela fonte “Arial” e a utilizará em todo documento. Se não encontrar, utilizada a “Helvetica”. Se não encontrar ambas, o navegador usará outra fonte da família sans-serif .



# CSS

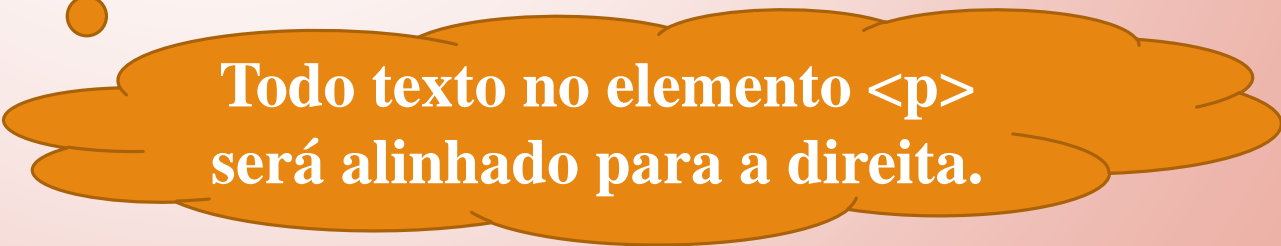
- Propriedade **font-style**: permite alterar o estilo da fonte no navegador. Temos:
  - normal: Fonte normal.
  - italic: Fonte itálica.
- Ex: **font-style: italic;**
- Propriedade **text-align**: Uma das propriedades mais simples, porém muito utilizada, que diz respeito ao alinhamento de texto.

# CSS

- **text-align** pode ter os seguintes valores:
  - **Center:** Texto alinhado no centro;
  - **Left:** Texto alinhado à esquerda (padrão);
  - **Right:** Texto alinhado à direita;
  - **Justify:** Texto justificado;



```
p {  
  text-align: right;  
}
```



Todo texto no elemento <p>  
será alinhado para a direita.

# CSS

- **Outras propriedades de espaçamento de texto:**

p {

line-height: 3px; /\* tamanho da altura de cada linha \*/

letter-spacing: 3px; /\* tamanho do espaço entre cada letra \*/

word-spacing: 5px; /\* tamanho do espaço entre cada palavra \*/

text-indent: 30px; /\* tamanho da margem da 1ª linha do texto \*/

}

# CSS

- Propriedade **background-image**: propriedade que permite selecionar um arquivo de imagem para ser exibido ao fundo do elemento selecionado.

```
h1 {  
    background-image: url(sobre-background.jpg);  
}
```

- O navegador vai requisitar o arquivo *sobre-background.jpg*, que deve estar na mesma pasta do arquivo CSS onde consta essa declaração.

# CSS

- Temos também propriedades para definirmos bordas para os elementos. Podemos alterar a cor, largura e estilo da borda, conforme abaixo:

```
body {  
  border-color: red; /* cor da borda */  
  border-style: solid; /* estilo de exibição da borda */  
  border-width: 1px; /* largura da borda */  
}
```

Outros estilos de borda:  
*dashed, dotted, double,*  
*groove.*

- Para que o efeito da cor sobre a borda surta efeito, é necessário que a propriedade *border-style* tenha qualquer valor diferente do padrão *none*.

# CSS

- Outras possibilidades para bordas

```
p{  
  border-bottom: 2px solid #333333; /* insere borda abaixo do elemento */  
}
```

- Teste e veja o resultado !!
- Para mais informações sobre propriedades do CSS:
  - <http://www.w3schools.com/css/>
  - <http://pt-br.learnlayout.com/clear.html>



# Sequência do Exercício: Primeira Página

- Aplicaremos um pouco de estilo em nossa página usando CSS (bordas, fontes, alinhamento de texto, imagens de fundo, etc).
- Dentro da pasta css, crie um arquivo sobre.css, que conterá nosso código de estilo para essa página.

## A Mirror Fashion

A Mirror Fashion é a maior empresa comércio eletrônico no segmento de moda em todo o mundo. Fundada em 1932, possui filiais em 124 países, sendo líder de mercado com mais de 90% de participação em 118 deles.

Nosso centro de distribuição fica em [Jacarezinho, no Paraná](#). De lá, saem 48 aviões que distribuem nossos produtos às casas do mundo todo. Nosso centro de distribuição:



Centro de distribuição da Mirror Fashion

Compre suas roupas e acessórios na Mirror Fashion. Acesse [nossa loja](#) ou entre em [contato](#) se tiver dúvidas. Conheça também nossa [história](#) e nossos [diferenciais](#).

### História

A fundação em 1932 ocorreu no momento da descoberta econômica do interior do Paraná. A *família Pelho*, tradicional da região, investiu todas as suas economias nessa nova iniciativa, revolucionária para a época. O fundador *Eduardo Simões Pelho*, dotado de particular visão administrativa, guiou os negócios da empresa durante mais de 50 anos, muitos deles ao lado de seu filho *E. S. Pelho Filho*, atual CEO. O nome da empresa é inspirado no nome da família.

O crescimento da empresa foi praticamente instantâneo. Nos primeiros 5 anos, já atendia 18 países. Bateu a marca de 100 países em apenas 15 anos de existência. Até hoje, já atendeu 740 milhões de usuários diferentes, em bilhões de diferentes pedidos.

O crescimento em número de funcionários é também assombroso. Hoje, é a maior empregadora do Brasil, mas mesmo após apenas 5 anos de sua existência, já possuía 30 mil funcionários. Fora do Brasil, há 240 mil funcionários, além dos 890 mil brasileiros nas instalações de Jacarezinho e nos escritórios em todo país.

Dada a importância econômica da empresa para o Brasil, a família Pelho já recebeu diversos prêmios, homenagens e condecorações. Todos os presidentes do Brasil já visitaram as instalações da Mirror Fashion, além de presidentes da União Européia, Ásia e o secretário-geral da ONU.



Família Pelho

### Diferenciais

- Menor preço do varejo, garantido
- Se você achar uma loja mais barata, leva o produto de graça
- Pague em reais, dólares, euros ou bitcoins
- Todas as compras com frete grátis para o mundo todo
- Maior comércio eletrônico de moda do mundo
- Atendimento via telefone, email, chat, twitter, facebook, carta, fax e telegrama
- Presente em 124 países
- Mais de um milhão de funcionários em todo o mundo

## Compre já em nossa loja!

# CSS

- Cores: É difícil acertar a exata variação de cor que queremos no design. Por isso, é bem incomum usarmos cores com seus nomes.
- O mais comum é definir a cor com base em sua composição RGB.

h3 {

background: #F2EDED;

}

Notação Hexadecimal

# CSS

- Propriedades de **espaçamento** e **margem**: para espaçamento temos a propriedade *padding* e para margem temos a propriedade *margin*.
- 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, sua borda, e seu respectivo conteúdo) e tem as subpropriedades listadas a seguir:
  - *padding-top*
  - *padding-right*
  - *padding-bottom*
  - *padding-left*
- Essas propriedades aplicam uma distância entre o limite do elemento e seu conteúdo acima, à direita, abaixo e à esquerda respectivamente.

# CSS

- Se passado somente um valor para o *padding*, o valor é aplicado em todas as direções.

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

- Se passados dois valores, o primeiro será aplicado acima e abaixo e o segundo será aplicado à direita e à esquerda.

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



# CSS

- Se passados três valores, o primeiro será aplicado acima, o segundo será aplicado à direita e à esquerda e o terceiro será aplicado abaixo do elemento

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

- Se passados quatro valores, serão aplicados respectivamente a padding-top, padding-right, padding-bottom e padding-left.

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



# CSS

- A propriedade *margin* é bem parecida com a propriedade *padding*, exceto que ela adiciona espaço após o limite do elemento, ou seja, é um espaçamento além do elemento em si.

Possui as seguintes subpropriedades:

- *margin-top*
  - *margin-right*
  - *margin-bottom*
  - *margin-left*
- Há ainda uma maneira de permitir que o navegador defina qual será a dimensão da propriedade *padding* ou *margin* conforme o espaço disponível na tela: definimos o valor auto para a margem ou o espaçamento que quisermos.

# CSS

- Dimensões: É possível determinar as dimensões de um elemento, por exemplo:

```
p {  
  background-color: red;  
  height: 300px; /* Altura do elemento */  
  width: 300px; /*Largura do elemento */  
}
```

- Todos os parágrafos do nosso HTML ocuparão 300 pixels de largura e de altura, com cor de fundo vermelha.

# CSS

- Links: para adicionar links em textos ou imagens, basta utilizar a tag de âncora <a>:

<a href="https://www.google.com">

Página principal do planeta !

</a>

Quando o usuário clicar  
no texto, ele será  
direcionado para a página  
inicial do google.

# CSS

- Outro uso para a tag de âncora é a demarcação de destinos para links dentro do próprio documento, o que chamamos de bookmark.

`<p>Mais informações <a href="#info">aqui</a>.</p>`

`<p>Conteúdo da página...</p>`

`<h2 id="info">Mais informações sobre o assunto:</h2>`

`<p>Informações...</p>`

- De acordo com o exemplo acima, ao clicarmos sobre a palavra "aqui", demarcada com um link, o usuário será levado à porção da página onde o bookmark "info" é visível. Bookmark é o elemento que tem o atributo id.

# CSS

- É possível, com o uso de um link, levar o usuário a um bookmark presente em outra página.

```
<a href="http://www.caelum.com.br/curso/wd43/#contato">
```

Entre em contato sobre o curso

```
</a>
```

- O exemplo acima fará com que o usuário que clicar no link seja levado à porção da página indicada no endereço, especificamente no ponto onde o bookmark "**contato**" seja visível.

# Sequência do Exercício: Primeira Página

- Aplicaremos margens em nossa página usando CSS. Também vamos criar links e listas.



# CSS

## Elementos Estruturais

- Já vimos muitas tags para casos específicos: **títulos com h1**, **parágrafos com p**, **imagens com img**, **listas com ul**, etc. E existem muitas outras além dessas !
- Você pode cair algum dia num cenário onde não consegue achar a tag certa para aquele conteúdo. Nesse caso, pode usar as tags **<div>** e **<span>** que funcionam como coringas.
- São tags sem nenhum significado especial mas que podem servir para agrupar um certo conteúdo, tanto um bloco da página quanto um pedaço de texto.

# CSS

## Seletores de ID e filho

- Às vezes não queremos pegar todos os parágrafos da página, mas apenas algum determinado. Para isso, utilizados **seletores de ID**.
- É possível aplicar propriedades visuais a um elemento selecionado pelo valor de seu **atributo id**. Para isso, o seletor deve iniciar com o caractere "#" seguido do valor correspondente.

# CSS

```
#cabecalho {  
  color: white;  
  text-align: center;  
}
```

- O seletor acima fará com que o elemento que tem o **id** com valor "**cabecalho**" tenha seu texto renderizado na cor branca e centralizado.
- Note que **não há indicação para qual tag a propriedade será aplicada**. Pode ser uma `<div>` ou um `<p>`, até mesmo tags sem conteúdo como uma `<img>`, desde que essa tenha o atributo `id` com o valor "**cabecalho**".

# CSS

- Podemos ainda utilizar um seletor hierárquico que permite aplicar estilos aos elementos filhos de um elemento pai:

```
#rodape img {  
  margin-right: 35px;  
  vertical-align: middle;  
  width: 94px;  
}
```

- No exemplo anterior, o elemento pai **rodape** é selecionado pelo seu id. O estilo será aplicado apenas nos elementos **img** filhos do elemento com **id=rodape**.



# CSS: Fluxo do Documento e Float

- Suponhamos que, por uma questão de design, a imagem da família Pelho deva vir ao lado do parágrafo e conforme a imagem abaixo:

## História

A fundação em 1932 ocorreu no momento da descoberta econômica do interior do Paraná. *A família Pelho*, tradicional da região, investiu todas as suas economias nessa nova iniciativa, revolucionária para a época. O fundador *Eduardo Simões Pelho*, dotado de particular visão administrativa, guiou os negócios da empresa durante mais de 50 anos, muitos deles ao lado de seu filho *E. S. Pelho Filho*, atual CEO. O nome da empresa é inspirado no nome da família.

O crescimento da empresa foi praticamente instantâneo. Nos primeiros 5 anos, já atendia 18 países. Bateu a marca de 100 países em apenas 15 anos de existência. Até hoje, já atendeu 740 milhões de usuários diferentes, em bilhões de diferentes pedidos.

O crescimento em número de funcionários é também assombroso. Hoje, é a maior empregadora do Brasil, mas mesmo após apenas 5 anos de sua existência, já possuía 30 mil funcionários. Fora do Brasil, há 240 mil funcionários, além dos 890 mil brasileiros nas instalações de Jacarezinho e nos escritórios em todo país.

Dada a importância econômica da empresa para o Brasil, a família Pelho já recebeu diversos prêmios, homenagens e condecorações. Todos os presidentes do Brasil já visitaram as instalações da Mirror Fashion, além de presidentes da União Européia, Ásia e o secretário-geral da ONU.



*Família Pelho*

# CSS: Fluxo do Documento e Float

- Isso não acontece por padrão. Observando as tags HTML que usamos até agora, os elementos da página **são desenhados um em cima do outro**.
- A tag <figure> ocupa toda largura da página e aparece empilhada no fluxo do doc., não permitindo que outros elementos sejam adicionados ao seu lado.
- Este problema pode ser solucionado por meio da propriedade **float**. Esta propriedade permite que tiremos um certo elemento do fluxo vertical do documento o que faz com que o conteúdo abaixo dele flua ao seu redor.



# Sequência do Exercício: Primeira Página

- Aplicaremos seletores CSS e flutuação de elementos em nossa página usando CSS.

# Princípios Básicos da Linguagem de Programação PHP.



# PHP

- Primeiramente, precisamos instalar o PHP na nossa máquina. Abra o seguinte link: [https://www.apachefriends.org/pt\\_br/download.html](https://www.apachefriends.org/pt_br/download.html)
- Escolha qualquer versão do Xampp e clique em baixar.
- O Xampp é uma distribuição do Apache fácil de instalar contendo PHP, MySQL, Perl, entre outros.

# PHP

- O XAMPP é um pacote com os principais servidores de código aberto do mercado, incluindo FTP, banco de dados MySQL e Apache com suporte as linguagens PHP e Perl.
- Como linguagem em servidores, usa-se PHP, Java, Ruby, ASP, .NET, Python, C#, SmallTalk, ColdFusion, etc.
- Alguns servidores: Apache, Tomcat (J2EE), Jboss (J2EE), IIS (Microsoft, com ASP e .NET), nginx e outros.



## XAMPP Control Panel v3.0.2

Config


Netstat

xampp-Shell

Explorer

Win-Services

Help

 Quit

## Modules

Service	Module	PID(s)	Port(s)	Action			
<input checked="" type="checkbox"/>	Apache			Start	Admin	Config	Logs
<input checked="" type="checkbox"/>	MySQL	2460	3306	Stop	Admin	Config	Logs
<input checked="" type="checkbox"/>	FileZilla			Start	Admin	Config	Logs
<input type="checkbox"/>	Mercury			Start	Admin	Config	Logs
<input type="checkbox"/>	Tomcat			Start	Admin	Config	Logs

```
11:43:10 [main] Xampp version: 1.7.7
11:43:10 [main] Control center version: 3.0.2 [ Compiled: Jul 21th 2011 - build #1 ]
11:43:10 [main] Running not as admin! This will work for all application stuff, but whenever you do
11:43:10 [main] something with services there will be a security dialogue! So think about running
11:43:10 [main] this application with administrator rights!
11:43:10 [main] Working with basedir: "c:\xampp\"
11:43:11 [main] Initializing moduls
11:43:11 [apache] "c:\xampp\apache\bin\httpd.exe" seems to be running on port 80?
11:43:11 [apache] "c:\xampp\apache\bin\httpd.exe" seems to be running on port 443?
11:43:11 [mysql] "c:\xampp\mysql\bin\mysqld.exe" seems to be running on port 3306?
11:43:11 [main] Starting check-timer
```

# PHP

- Usamos uma linguagem no servidor para executar tarefas como:
  - Gerar páginas dinamicamente com dados de um banco de dados da aplicação;
  - Enviar emails para usuários;
  - Processar tarefas complexas;
  - Garantir validações de segurança da aplicação;
  - Outras funções específicas.



# PHP

- Se escrevermos um HTML simples num arquivo .html, basta abri-lo no navegador e pronto, já podemos visualizar a página. Mas quando envolvemos um servidor o processo não é tão simples.
- Usamos o protocolo HTTP para servir páginas na Web. É por isso que todo endereço na Web começa com http://.
- Quando acessamos um endereço desses na Internet, estamos fazendo uma requisição ao servidor. Ou seja, pedimos que certo conteúdo seja exibido.

# PHP

- O servidor pode executar código e gerar HTML na hora pro cliente, dinamicamente. É esse processamento de lógica dinâmica no servidor que queremos fazer com PHP.
- Para executar código do lado do servidor com PHP, basta renomear o arquivo de .html para .php.
- Com essa extensão, podemos agora misturar código dinâmico PHP no meio do nosso HTML

# Primeiro exemplo com PHP

- Teste: instale o Xampp na pasta do disco rígido C:/...
- Abra o arquivo xampp-control.exe
- Aperte start nos módulos “apache” e “mySQL”.
- Agora poderemos testar nossas aplicações desenvolvidas em linguagem PHP !!

# Primeiro exemplo com PHP

- Teste: Na pasta onde vocês instalaram o xampp, no diretório **htdocs** (C:\xampp\htdocs), criem um arquivo teste.php;
- Copiem e cole o seguinte trecho de código e salvem o arquivo:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Teste</title>
  </head>
  <body>
    <p>
      <?php echo "Olá, sou um script PHP !!" ?>
    </p>
  </body>
</html>
```

# Primeiro exemplo com PHP

- Teste: Agora abram algum navegador de preferência com o seguinte endereço:
- <http://localhost:85/teste.php>
- A instrução **echo** do PHP permite mostrar mensagens na tela, similar ao printf da linguagem C e ao system.out.println da linguagem JAVA.
- O diretório raiz do nosso servidor web será o endereço: C:\xampp\htdocs, ou seja, nossas aplicações PHP deverão ser desenvolvidas nessa pasta.

# PHP

- O PHP (PHP: Hypertext Preprocessor) é uma linguagem open source de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do HTML.
- O código PHP é delimitado pelas instruções de processamento (tags) de início e fim `<?php` e `?>` que permitem que você pule para dentro e para fora do "modo PHP".



# PHP

- O que distingue o PHP do JavaScript é que o código é executado no servidor, gerando o HTML que é então enviado para o navegador.
- O navegador recebe os resultados da execução desse script, mas não sabe qual era o código fonte.
- Você pode configurar seu servidor web para processar todos os seus arquivos HTML com o PHP, e então não haverá nenhum modo dos usuários descobrirem se você usa essa linguagem ou não.

# PHP

- O código é interpretado no lado do servidor pelo módulo PHP, que também gera a página web a ser visualizada no lado do cliente.
- Características:
  - Velocidade e robustez.
  - Orientação a objetos.
  - Portabilidade - independência de plataforma - escreva uma vez, rode em qualquer lugar.
  - Tipagem dinâmica.
  - Sintaxe similar a C/C++ e o Perl.
  - Open-source (código aberto).
  - Server-side (O cliente manda o pedido e o servidor responde em página HTML).



# Sintaxe da linguagem PHP

# Sintaxe do PHP

- Como vimos anteriormente, a função “echo” exibe strings (Uma string é uma palavra ou frase entre aspas, assim: "Olá, mundo!").

- Exemplo:

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <?php
      echo "Estou aprendendo PHP";
    ?>
  </body>
</html>
```

# Sintaxe do PHP

- Temos o operador de concatenação ( . ) que une strings. Observe abaixo:

- Exemplo:

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <h1>
      <?php
        echo "Estou" . " aprendendo PHP";
      ?>
    </h1>
  </body>
</html>
```

# Sintaxe do PHP

- Podemos também fazer cálculos aritméticos diretamente na instrução “echo”:

- Exemplo:

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <h1>
      <?php
        echo 12*5;
      ?>
    </h1>
  </body>
</html>
```



# Sintaxe do PHP

- Variáveis: Uma variável pode armazenar uma string ou número, e dá ao dado armazenado um nome específico. O PHP é **Case Sensitive**:

```
<?php  
$myName = "Luciano";  
$myAge = 23;  
echo $myName;  
echo $myAge;  
?>
```

Porque não declaramos o  
tipo das variáveis?

Tipagem Dinâmica !!

# Sintaxe do PHP

- Para fazer comentários em PHP, basta usar barras duplas //.
- Exemplo:

```
<?php  
    echo 12*5; //Trecho que faz um cálculo aritmético  
?>
```

- O PHP possui os seguintes **operadores de comparação**: **>, <, >=, <=, ==, !=;**
- **Operações matemáticas**: **+, -, \*, /, % (módulo), \*\* (exponencial);**
- **Operadores Lógicos**: **and, or, &&, ||, !, xor;**

# Sintaxe do PHP

- Temos também estruturas de condição: IF-ELSE.
- Exemplo:

**<?php**

```
$myAge = 23;  
if ($myAge > 20){  
    echo “Você é Idoso”;  
}  
elseif($myAge > 40){  
    echo “Você é uma múmia”;  
}  
else{  
    echo “Você está na flor da idade”;  
}
```

**?>**

# Sintaxe do PHP


- Temos também a estrutura condicional SWITCH:

- Exemplo:

```
<?php
$fruit = "Apple";
switch ($fruit) {
    case 'Apple':
        echo "Yummy.";
        break;
    default:
        echo "Nenhuma fruta";
}
?>
```

# Sintaxe do PHP

- Quando declaramos um array, temos que usar array(). Basicamente, isso diz ao PHP que \$array é um array, e não algo como uma velha e comum variável.
- Cada item em um array deve ser separado por uma vírgula.



```
<?php
```

```
$array = array("Egg", "Tomato", "Beans", "Chips", "Sausage" );
```

```
?>
```

# Sintaxe do PHP

- Cada item em um array é numerado a partir de 0, igual a linguagem C:

```
<?php
```

```
$tens = array(10, 20, 30, 40, 50);
```

```
echo $tens[2];
```

```
$languages = array("HTML/CSS","JavaScript","PHP","Python","Ruby");
```

```
$languages[1] = "C";
```

```
unset($languages[4]);
```

```
echo $languages[1];
```

```
?>
```

- Você também pode usar “**unset**” para remover itens do array.
- Obs: Pode ser usado colchetes ou chaves para definir o índice a ser acessado.



# Sintaxe do PHP

- Estruturas de Repetição: **FOR**;

```
<?php
```

```
    for ($leap = 2004; $leap < 2050; $leap = $leap + 4) {  
        echo "$leap";  
    }
```

```
?>
```

# Sintaxe do PHP

- O laço **FOREACH** é usado para iterar por cada elemento de um objeto — o que faz com que ele seja perfeito para ser usado com arrays!

```
<?php
```

```
$sentence = array("I'm ", "learning ", "PHP!");
```

```
foreach ($sentence as $word) {
```

```
    echo $word;
```

```
}
```

```
?>
```

# Sintaxe do PHP

- Temos também as estruturas de repetição While e Do.While:

```
<?php
```

```
$loopCond = true;  
while ($loopCond){  
    echo "O laço está em execução.";  
    $loopCond = false;  
}  
echo "E agora está terminado.";  
do{  
    echo "O laço está em execução.";  
}  
while($loopCond);
```

```
?>
```

# Sintaxe do PHP

- Algumas Funções nativas do PHP:
  - **strlen**: Passando uma string como parâmetro, a função retorna o n° de caracteres da mesma (**\$tam = strlen("Luciano");**);
  - **substr**: Permite que você retorne uma substring (um pedaço) da sua string.  
`$myname = "David";`  
`$partial = substr($myname, 0, 3);`  
`print $partial; //Imprime Dav`
  - **strtoupper** e **strtolower**: Torna todos caracteres da string maiúsculos e minúsculos, respectivamente.

# Sintaxe do PHP

- Algumas Funções nativas do PHP:
  - **strpos**: encontra a posição da primeira ocorrência de uma substring em uma string;  

```
$pos = strpos("emily", "e"); // 0  
$pos = strpos("emily", "i"); // 2  
$pos = strpos("emily", "ily"); // 2  
$pos = strpos("emily", "zxc"); // falso
```
  - **array\_push**: Recebe dois argumentos: um array, e um elemento para adicionar no final deste array.
  - **count**: Conta os elementos de um array.
  - **sort** e **rsort**: Ordena e ordena na ordem inversa os elementos de um array.

# Exercício com PHP

- **Exercício exemplo com PHP , HTML e CSS: Vamos criar uma página que receba do usuário seu nome, idade, altura e peso.**
- A página deve mostrar se o usuário está acima, abaixo ou no peso ideal.
- Fórmula do peso ideal:  $\text{peso}/\text{altura}*\text{altura}$ .
- A página deve mostrar também a tabela ao lado.
- A página deve mostrar em qual categoria o usuário está.

Abaixo do peso	Abaixo de 18.4
<b>Peso normal</b>	<b>18.5 - 24.9</b>
Sobrepeso	25.0 - 29.9
Obesidade Grau I	30.0 - 34.9
Obesidade Grau II	35.0 - 39.9
Obesidade Grau III	Acima de 40



# Exercício com PHP

- 1º Passo: Receber as informações do usuário com formulários HTML.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<form action="action.php" method="POST">
```

```
  Digite seu nome:<br>
```

```
  <input type="text" name="nome" value="Mickael">
```

```
  <br>
```

```
  Digite sua idade:<br>
```

```
  <input type="text" name="idade" value="50">
```

```
  <br>
```

```
  Digite sua altura:<br>
```

```
  <input type="text" name="altura" value="1.81">
```

```
  <br>
```

```
  Digite seu peso:<br>
```

```
  <input type="text" name="peso" value="80">
```

```
  <br>
```

```
  <input type="submit" value="Submit">
```

```
</form>
```

```
</body>
```

```
</html>
```

O atributo **action** especifica onde enviar os dados do formulário quando um formulário é enviado.

O atributo **value** pré-define um valor.

O atributo **name** serve para "empacotar" os dados informados e enviá-los para uma outra página.

Entrada de dados tipo **“texto”**

Entrada de dados através do botão **“submit”**.

# Exercício com PHP

- 2º Passo: Vamos aplicar um pouco de estilo na página utilizando CSS. Criem um arquivo style.css no mesmo diretório e copiem o código a seguir:

```
body{
    margin-left: auto;
    margin-right: auto;
    width: 900px;
}
#tabela{
    float: right;
    width: 400px;
}
#inputs {
    width: 400px;
    background-color: #F2EDED;
    border: 3px solid #ccc;
    text-align: center;
}
```

- 2º Passo: Mude o que está em vermelho no imc.html:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <link rel="stylesheet" href="style.css">
```

```
  </head>
```

```
<body>
```

```
  <div id="tabela">
```

```
    <table border="1">
```

```
      <tr><td>Categoria</td> <td> IMC </td> </tr>
```

```
      <tr><td>Abaixo do Peso Ideal</td> <td> Abaixo de 18.4 </td> </tr>
```

```
      <tr><td>Peso Ideal</td> <td> 18.5 - 24.9 </td> </tr>
```

```
      <tr><td>Sobrepeso</td> <td> 25 - 29.9 </td> </tr>
```

```
      <tr><td>Obesidade grau I</td> <td> 30 - 34.9 </td> </tr>
```

```
      <tr><td>Obesidade grau II</td> <td> 35 - 39.9 </td> </tr>
```

```
      <tr><td>Obesidade grau III</td> <td> 40 ou mais </td> </tr>
```

```
    </table>
```

```
  </div>
```

```
  <div id="inputs">
```

- 2º Passo: Mude o que está em vermelho no imc.html:

```
<form action="action_page.php" method="POST">
```

```
  Digite seu nome:<br>
```

```
  <input type="text" name="nome" value="Mickey">
```

```
  <br>
```

```
  Digite sua idade:<br>
```

```
  <input type="text" name="idade" value="50">
```

```
  <br>
```

```
  Digite sua altura:<br>
```

```
  <input type="text" name="altura" value="1.81">
```

```
  <br>
```

```
  Digite seu peso:<br>
```

```
  <input type="text" name="peso" value="100">
```

```
  <br>
```

```
  <input type="submit" value="Calcula<b>r</b>">
```

```
</form>
```

```
<b></div>
```

```
</body>
```

```
</html>
```

# Exercício com PHP

- 3º Passo: Vamos criar um arquivo chamado **action\_page.php** **no mesmo diretório**. Copiem o código a seguir:

```
<html>
```

```
<head>
```

```
<title> <?php echo $_POST['nome']; ?> </title>
```

```
</head>
```

```
<body>
```

```
<?php
```

```
$nome = $_POST['nome'];
```

```
$idade = $_POST['idade'];
```

```
$altura = $_POST['altura'];
```

```
$peso = $_POST['peso'];
```

```
$imc = $peso/($altura*$altura);
```

O método `$_POST`  
captura os dados do  
elemento com name  
= “xxx”

# Exercício com PHP

- 3º Passo: Copiem o código a seguir:

```
if($imc<18.4){
    echo "Olá ".$nome."!." "Você está abaixo do peso ideal com IMC ".$imc;
}
elseif($imc<24.9){
    echo "Olá ".$nome."!." "Você está no peso ideal com IMC ".$imc;
}
elseif($imc<29.9){
    echo "Olá ".$nome."!." "Você está com sobrepeso com IMC ".$imc;
}
elseif($imc<34.9){
    echo "Olá ".$nome."!." "Você está no grau de obesidade I com IMC ".$imc;
}
elseif($imc<39.9){
    echo "Olá ".$nome."!." "Você está no grau de obesidade II com IMC ".$imc;
}
else{
    echo "Olá ".$nome."!." "Você está no grau de obesidade III com IMC ".$imc;
}
?>
```



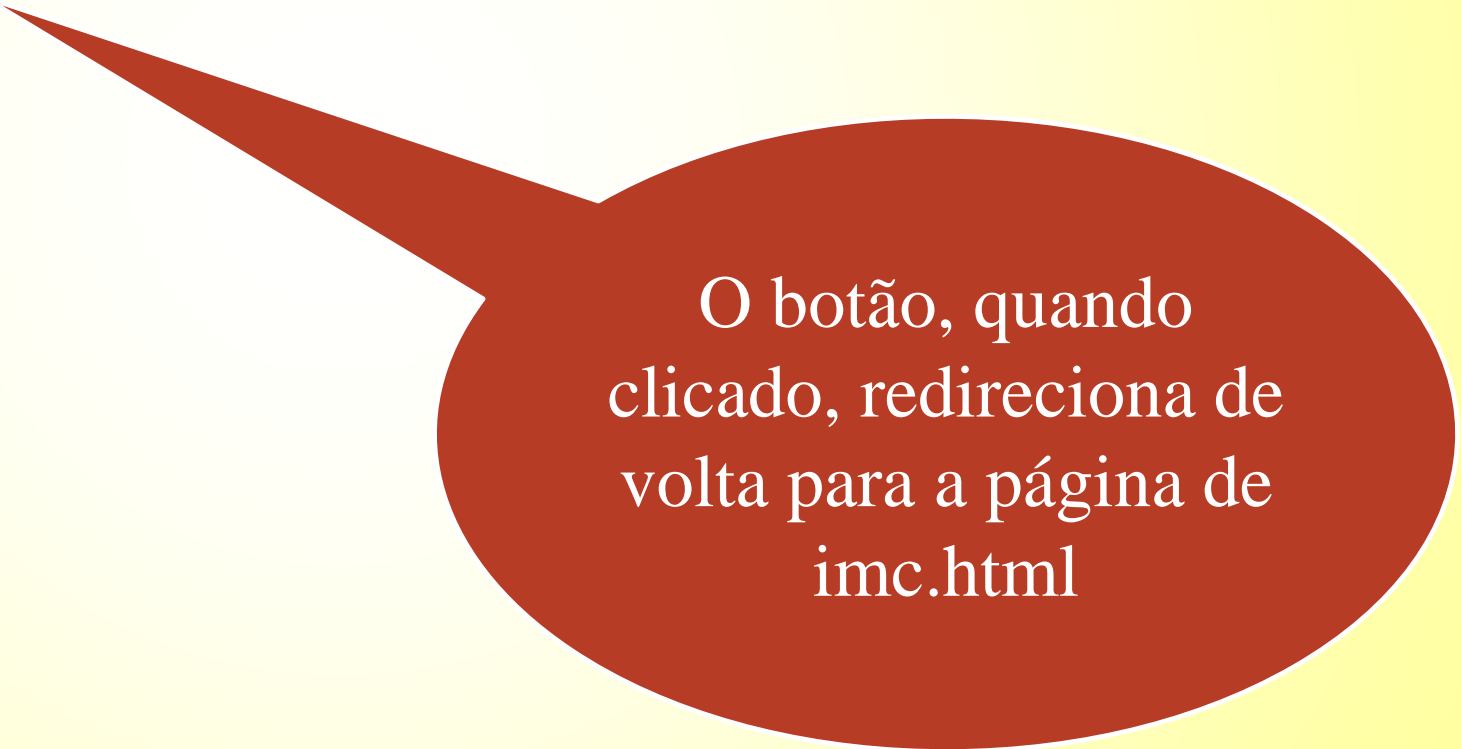
# Exercício com PHP

- 3º Passo: Copiem o código a seguir:<html>

```
<form method="POST" action="imc.html">  
<input type="submit" value="Voltar">  
</form>
```

```
</body>
```

```
</html>
```



O botão, quando clicado, redireciona de volta para a página de imc.html

Digite seu nome:

Luciano

Digite sua idade:

23

Digite sua altura:

1.81

Digite seu peso:

100

Calcular

Categoria	IMC
Abaixo do Peso Ideal	Abaixo de 18.4
Peso Ideal	18.5 - 24.9
Sobrepeso	25 - 29.9
Obesidade grau I	30 - 34.9
Obesidade grau II	35 - 39.9
Obesidade grau III	40 ou mais

Olá Luciano! Você está no grau de obesidade I com IMC =30.524098775984

[Voltar](#)

# Princípios Básicos da Linguagem de Programação JavaScript.



# JavaScript

- O JavaScript é uma linguagem de scripting, ou seja, é definida como uma linguagem de programação que permite ao programador controlar uma ou mais aplicações de terceiros.
- No caso do JavaScript, podemos controlar alguns comportamentos dos navegadores através de trechos de código que são enviados na página HTML.

# JavaScript

- Outra característica comum nas linguagens de scripting é que normalmente elas são linguagens interpretadas, ou seja, não dependem de compilação para serem executadas.
- Essa característica é presente no JavaScript: o código é interpretado e executado conforme é lido pelo navegador, linha a linha, assim como o HTML.




# JavaScript

- Porque JavaScript?
- O JavaScript pode controlar o comportamento do navegador em diversos aspectos;
- É a linguagem de programação do lado do cliente (Client-side) mais utilizada;
- Permite inserir vários efeitos, fazendo com que o site fique mais dinâmico;
- Permite executar instruções como resposta às ações do usuário;

# JavaScript

- Relembrando: O PHP roda no **servidor** e o JS roda no **cliente**.
- Para rodar JavaScript em uma página Web, precisamos ter em mente que a execução do código é instantânea. Para inserirmos um código JavaScript em uma página, é necessário utilizar a tag **<script>**:



```
<script>  
  alert("Olá, Mundo!");  
</script>
```

- O exemplo acima é um "hello world" em JavaScript utilizando uma função do navegador, a função alert.

# JavaScript

- A tag `<script>` pode ser declarada dentro da tag `<head>` assim como na tag `<body>`, mas devemos ficar atentos, porque o código é lido imediatamente dentro do navegador.
- Devemos sempre colocar o script antes de fecharmos a tag `</body>`? **Na maioria esmagadora das vezes sim.**
- Porquê?

# JavaScript

- Imagine ter que escrever o script toda vez que ele fosse necessário em outra página? Por isso é possível importar scripts dentro da página utilizando a tag `<script>`. No arquivo HTML:
  - `<script src="hello.js"></script>`
- No arquivo externo **hello.js**:
  - `alert("Olá, Mundo!");`
- Com a separação do script em arquivo externo é possível reaproveitar alguma funcionalidade em mais de uma página.

# JavaScript

- Observe o exemplo abaixo. Salve o arquivo como lampada.html.

```
<!DOCTYPE html>
```

```
<html>
```

```
  <body>
```

```
    <h1>What Can JavaScript Do?</h1>
```

```
    <p>JavaScript can change HTML attributes.</p>
```

```
    <p>In this case JavaScript changes the src (source) attribute of an image.</p>
```

```
    <button
```

```
      onclick='document.getElementById('myImage').src='pic_bulbon.gif''>Turn  
on the light</button>
```

```
    
```

```
    <button
```

```
      onclick='document.getElementById('myImage').src='pic_bulboff.gif''>Turn  
off the light</button>
```

```
  </body>
```

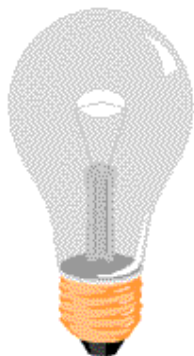
```
</html>
```



## What Can JavaScript Do?

JavaScript can change HTML attributes.

In this case JavaScript changes the src (source) attribute of an image.



Turn on the light

Turn off the light



# JavaScript

- Observe o exemplo abaixo. Salve o arquivo como lampada.html.

```
<!DOCTYPE html>
```

```
<html>
```

```
  <body>
```

```
    <h1 id="i1">What Can JavaScript Do?</h1>
```

```
    <p id="i2">JavaScript can change HTML attributes.</p>
```

```
    <p id="i3">In this case JavaScript changes the src (source) attribute of an image.</p>
```

```
    <button onclick="document.getElementById('myImage').src='pic_bulbon.gif'">Turn on the  
light</button>
```

```
    
```

```
    <button onclick="document.getElementById('myImage').src='pic_bulboff.gif'">Turn off the  
light</button>
```

```
    <button onclick="document.getElementById('i1').style.color = 'blue';  
document.getElementById('i2').style.fontFamily = 'Arial';  
document.getElementById('i3').style.fontSize = '50px';">Mudar Texto</button>
```

```
  </body>
```

```
</html>
```

# JavaScript

- Assim como no PHP, O JS também possui estruturas de condição, seleção, repetição e funções;
- O formato e a sintaxe são similares às de linguagens já estudadas (C, PHP, ...);
- A seguir, vamos utilizar JavaScript para tornar nossa página sobre.html mais dinâmica.

# JavaScript : exercício

- Vamos alterar novamente nossa página sobre.html;
- Na tag <head>, <img> e <Strong> e insiram o seguinte trecho:  

```
<script src="js/java.js"></script>  
  
<strong id="mf" onmouseout="funcao3('mf',0)" >  
<h3 onmouseover="funcao4('h3')" onmouseout="funcao5('h3')">Compre já  
em nossa Loja!</h3>
```
- Feito isso, adicione o seguinte no arquivo sobre.css:  

```
#header {  
    width: 150px;  
    height: 100px;  
    position: relative;  
}
```

# JavaScript : exercício

- Agora vamos criar o arquivo `java.js` no diretório JS;
- Coloquem o código que será exibido no quadro.
- Salvem o arquivo e abram a página `sobre.html` e vejam alguns dos possíveis efeitos.



# Dúvidas ??

Contato: [lucianobrum18@gmail.com](mailto:lucianobrum18@gmail.com)



# Referências

- **Introdução a HTML e CSS.** Disponível em:

<https://www.caelum.com.br/apostila-html-css-javascript/introducao-a-html-e-css/>

- **W3Schools.** Disponível em: <http://www.w3schools.com/>