



Insper

Tecnologias Web

Aula 02 – Desafio CSS



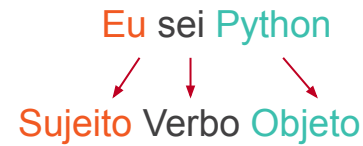
Antes de continuar...

Vamos fazer uma discussão sobre a aula passada

Aquecimento: LINGUAGENS DE PROGRAMAÇÃO

Sintaxe

- Regras para “funcionar”



Semântica

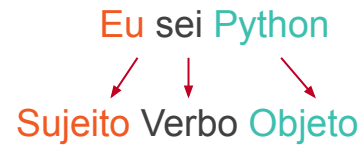
- O que é provocado / significado

Tradução

- Compilar ou interpretar

Sintaxe

- Regras para “funcionar”



Semântica

- O que é provocado / significado

Tradução

- Compilar ou interpretar

```
self.file = None
self.fingerprints = self()
self.logdups = True
self.debug = debug
self.logger = logging.getLogger(__name__)
if path:
    self.file = open(os.path.join(path, 'fingerprint.log'), 'a')
    self.file.seek(0)
    self.fingerprints.update({path: self.file})

    @classmethod
    def from_settings(cls, settings):
        debug = settings.getbool('debug')
        return cls(job_dir(settings), debug)

    def request_seen(self, request):
        fp = self.request_fingerprint(request)
        if fp in self.fingerprints:
            return True
        self.fingerprints.add(fp)
        if self.file:
            self.file.write(fp + os.linesep)

    def request_fingerprint(self, request):
        return request_fingerprint(request)
```

Tipos de linguagem

- A. Linguagem Procedural
- B. Linguagem de Máquina
- C. Linguagem Funcional
- D. Linguagem Orientada a Objetos
- E. Linguagem de Marcação

A. Linguagem Procedural / Imperativa

← Atribuição

Laços

↗ Condição

Função

```
def add5(x):  
    return x+5  
  
def dotwrite(ast):  
    nodename = getNodeName()  
    label=symbol.sym_name.get(int(ast[0]), ast[0])  
    print '    %s [label="%s" % (nodename, label),  
    if isinstance(ast[1], str):  
        if ast[1].strip():  
            print '= %s';' % ast[1]  
        else:  
            print '['  
    else:  
        print '];'  
    children = []  
    for n, child in enumerate(ast[1:]):  
        children.append(dotwrite(child))  
    print '    %s -> {' % nodename,  
    for name in children:  
        print '%s' % name,
```

C
Pascal
Cobol
Fortran
BASIC
...

B. Linguagem de Máquina

Linguagem de máquina

```
101101101  
111001101  
110001000
```

Linguagem de baixo nível

```
LEAD BASE  
ADD BONUS  
STORE SALARIO
```

Linguagem de alto nível

```
SALARIO = BASE + BONUS
```

B. Linguagem de Máquina

```
ORG 100H

MOV DX,STRING
MOV AH,9
INT 21H
MOV AH,4CH

INT 21H

STRING DB 'OLÁ, MUNDO!',0DH,0AH,'$'
```

ASSEMBLY

Mostra na tela a frase:
OLÁ, MUNDO!

C. Linguagem Funcional

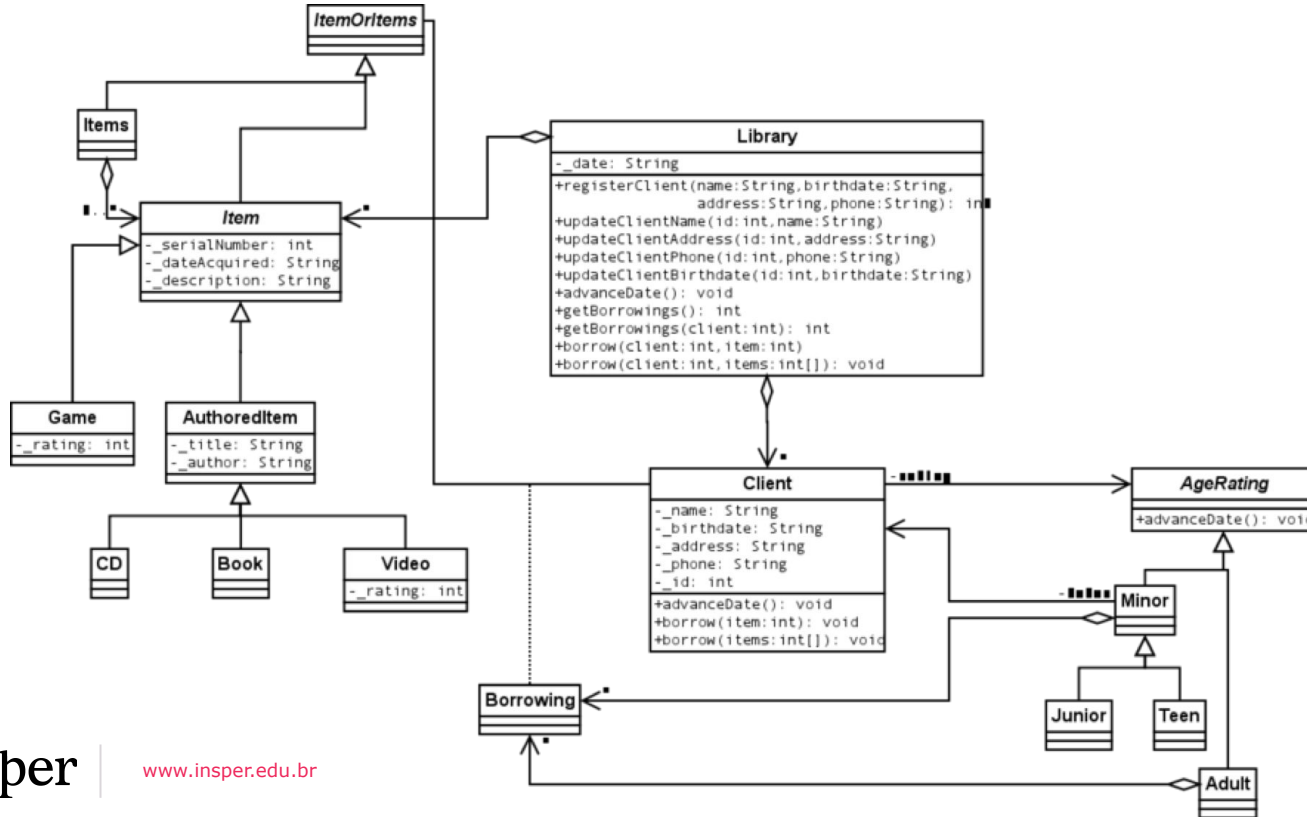
Baseado em aplicação de funções

Evita estados ou dados mutáveis

Não há alocação explícita de memória,
nem declaração explícita de variáveis

Scala
Erlang
LISP
...

D. Linguagem Orientada a Objetos



Java

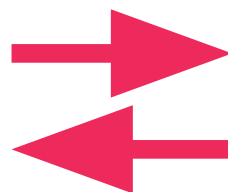
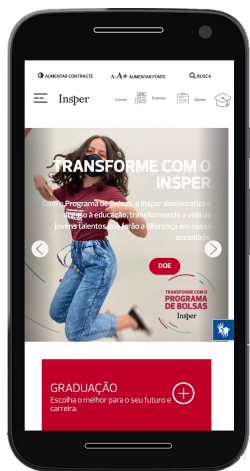
C#

C++

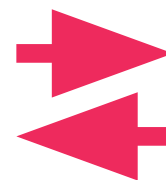
...

E. Linguagem de Marcação

FRONT-END

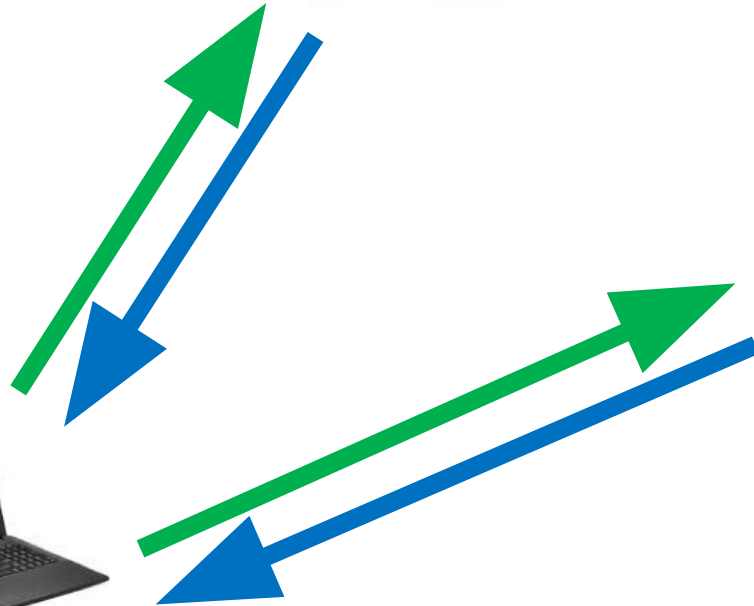
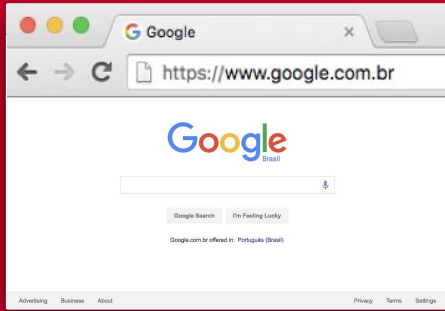


BACK-END



DNS

google.com.br → 172.217.28.131



Conclusão

WWW = World Wide Web

Sistema em hipermídia (várias mídias **interligadas por sistemas eletrônicos de comunicação** e executadas na **Internet**).

Três parâmetros da Web:

- **URL**
- **HTTP** (protocolo de comunicação)
- **HTML** (codificação da informação)

HTTP (Hypertext Transfer Protocol)

Protocolo = padrão de escrita para que periféricos se entendam.

O navegador prepara uma carta para o servidor onde fica o site da URL que você digitou

```
GET / HTTP/1.1
Host: www.google.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; pt-BR; rv:1.9.0.6) Gecko/2009011913
Firefox/3.0.6 (.NET CLR 3.5.30729)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: pt-br,pt;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
```

```
Location: http://www.google.com.br/
Cache-Control: private
Content-Type: text/html; charset=UTF-8
Date: Mon, 08 Feb 2009 22:00:00 GMT
Server: gws
Content-Length: 222
<CONTEÚDO DO SITE>
```

HTTP
Request

HTTP
Response

HTTPS?

HTTPS (Hypertext Transfer Protocol Secure)

Inserir uma camada de proteção na transmissão de dados entre seu computador e o servidor. Em sites com endereço HTTPS, a comunicação é criptografada, aumentando significativamente a segurança dos dados.

Infelizmente, não há HTTPS em todo lugar. O usuário, na verdade, depende que os sites ofereçam suporte a esse tipo de conexão para poder aproveitar da codificação. O que é importante ter em mente é que qualquer serviço online no qual seja necessário digitar uma senha para logar ou, principalmente, enviar dados de cartão de crédito, precisa de conexão via HTTPS

Verbos HTTP

Métodos de requisição

POST

Criar novos recursos. Na criação bem-sucedida, retornar o status [HTTP 201](#).

GET

Ler/recuperar uma representação de um recurso. Em caso de sucesso, retorna o recurso e um código de resposta [HTTP 200 \(OK\)](#). Em caso de erro, retorna [404 \(NOT FOUND\)](#) ou [400 \(BAD REQUEST\)](#).

PUT

Substituir/atualizar recursos, em uma URI de recurso conhecido, com a representação atualizada no corpo da requisição. Em caso de sucesso, retorna [200](#) (ou [204](#) se não retornar conteúdo no corpo).

PATCH

Modificar parcialmente os recursos. A requisição só precisa conter as alterações específicas para o recurso, não o recurso completo.

DELETE

Excluir um recurso identificado por uma URI. Na exclusão bem-sucedida, devolve o status [HTTP 200 \(OK\)](#) ou o status [HTTP 204 \(NO CONTENT\)](#) sem corpo de resposta.

Outros protocolos?

TCP/IP = PROTOCOLO DE CONTROLE DE TRANSMISSÃO / PROTOCOLO INTERNET

Forma padronizada de comunicação para os computadores na internet. O TCP faz a divisão dos dados que será enviada em segmentos de dados.

FTP = FILE TRANSFER PROTOCOL

Recurso utilizado para transferências de arquivos pela internet. Este protocolo permite apenas o envio e o recebimento dos arquivos, no entanto para a leitura serão necessários outros programas.

SSL = SECURE SOCKETS LAYER

Este é um protocolo de segurança que permite a confirmação da identidade de um servidor, verificando o nível de confiança.

ICMP = INTERNET CONTROL MESSAGE PROTOCOL

Autoriza a criação de mensagens relativas ao IP, mensagens de erro e pacotes de teste.

SMTP = SIMPLE MAIL TRANSFER PROTOCOL

Usado para envios de mensagem em rede funcionando como roteador do correio eletrônico.

IMAP = INTERNET MESSAGE ACCESS PROTOCOL

Permite a manipulação de caixas postais remotas como se fossem locais, permitindo a organização da forma que melhor convier.

POP = POST OFFICE PROTOCOL

Usado para receber mensagens de e-mail. Permitem apenas o acesso a uma caixa de e-mail, já o IMAP permite o acesso a várias.

MIME = MULTIPURPOSE INTERNET MAIL EXTENSIONS

Permite ao usuário a utilização do e-mail trabalhando em multimídia.

Desafio CSS

Lembra de Co-Design de Aplicativos? Não? Está na hora de lembrar!

Aquecimento: LINGUAGENS DE MARCAÇÃO

HTML

```
<h1>Título</h1>
```

```
<h2>Subtítulo</h2>
```

```
<p><strong>Lorem Ipsum</strong> é simplesmente um texto fictício.
```

Markdown

`# Título`

`## Subtítulo`

`**Lorem Ipsum** é simplesmente um texto fictício.`

Texto Renderizado

Título

Subtítulo

Lorem Ipsum é simplesmente um texto fictício.

Linguagem de Marcação

HTML/CSS

Markdown

Latex

XML

Linguagem de Programação

Python

Java

JavaScript

C



Quais as
diferenças entre
as duas?

Desafio CSS

Lembra de Co-Design de Aplicativos? Não? Está na hora de lembrar!

CSS

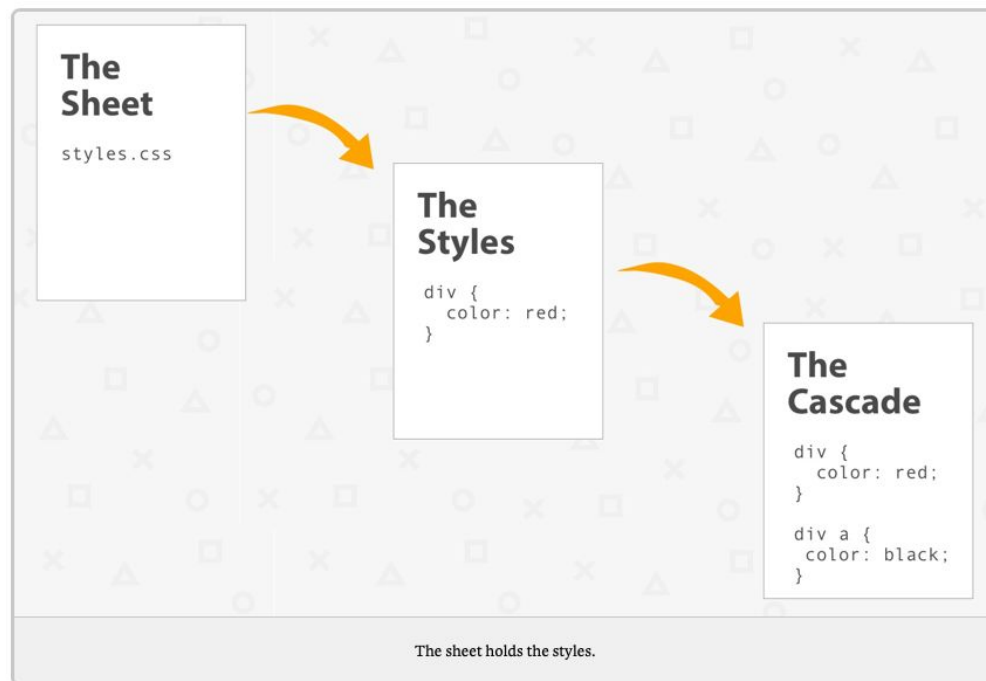


Imagem retirada de: <https://css-tricks.com/css-basics-second-s-css/>

The Sheet: FORMAS DE UTILIZAÇÃO DO CSS

Inline

```
<html>  
  <body>  
    <h1 style="background-color: red;">Hello world!</h1>  
  </body>  
</html>
```

Style Block

```
<html>
  <head>
    <style>
      h1{
        background-color: red;
      }
    </style>
  </head>
  <body>
    <h1>Hello world!</h1>
  </body>
</html>
```

Arquivo Externo

```
<html>  
  <head>  
    <link rel="stylesheet" href="file.css">  
  </head>  
  <body>  
    <h1>Hello world!</h1>  
  </body>  
</html>
```

file.css

```
h1{  
  background-color:red;  
}
```

The Style

Seletores do CSS

Elemento
HTML

```
h1{  
  color: blue;  
}
```

Classe

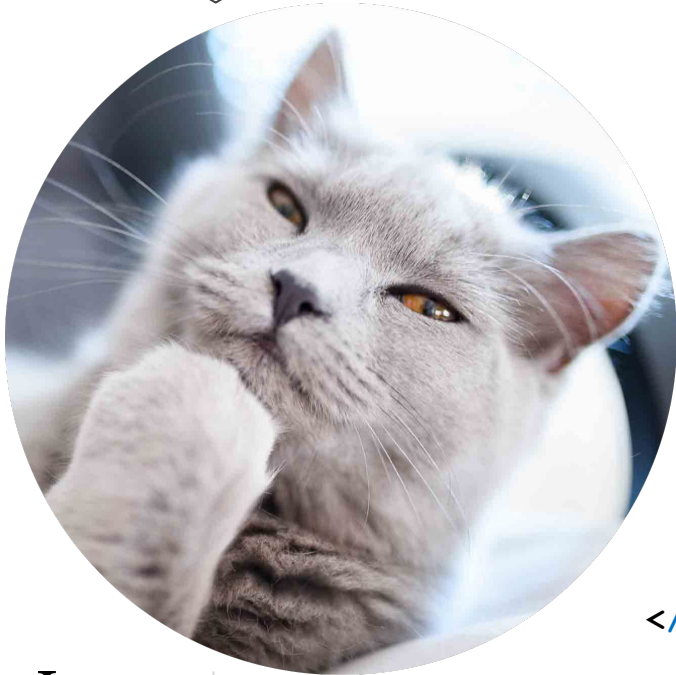
```
.nome-da-classe{  
  color: blue;  
}
```

ID

```
#algum-id{  
  color: blue;  
}
```

The Cascade

O elemento `<h1>` terá qual cor no plano de fundo?

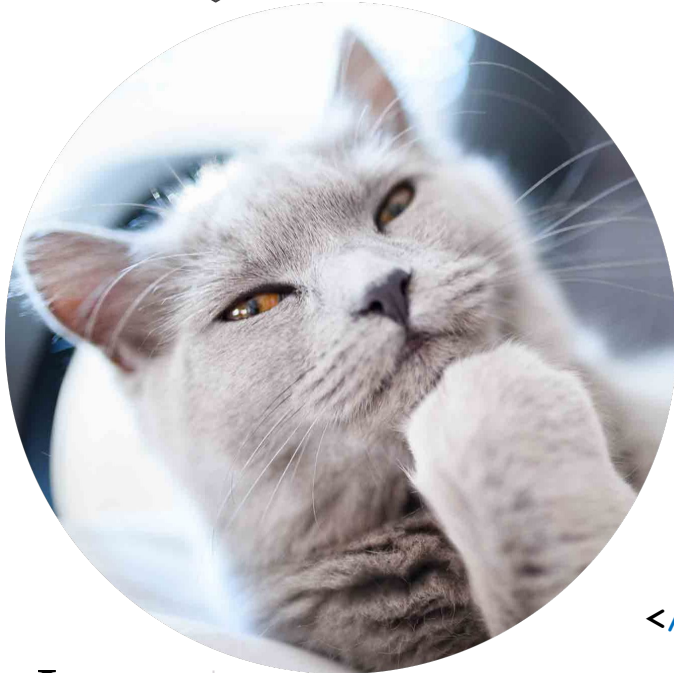


```
<html>
  <head>
    <style>
      h1{
        background-color: black;
      }

      .classe{
        background-color: pink;
      }

      #id{
        background-color: purple;
      }
    </style>
  </head>
  <body>
    <h1 class="classe" id="id" style="background-color:blue;">
      Hello world!
    </h1>
  </body>
</html>
```

Será que tem alguma ordem de prioridade?

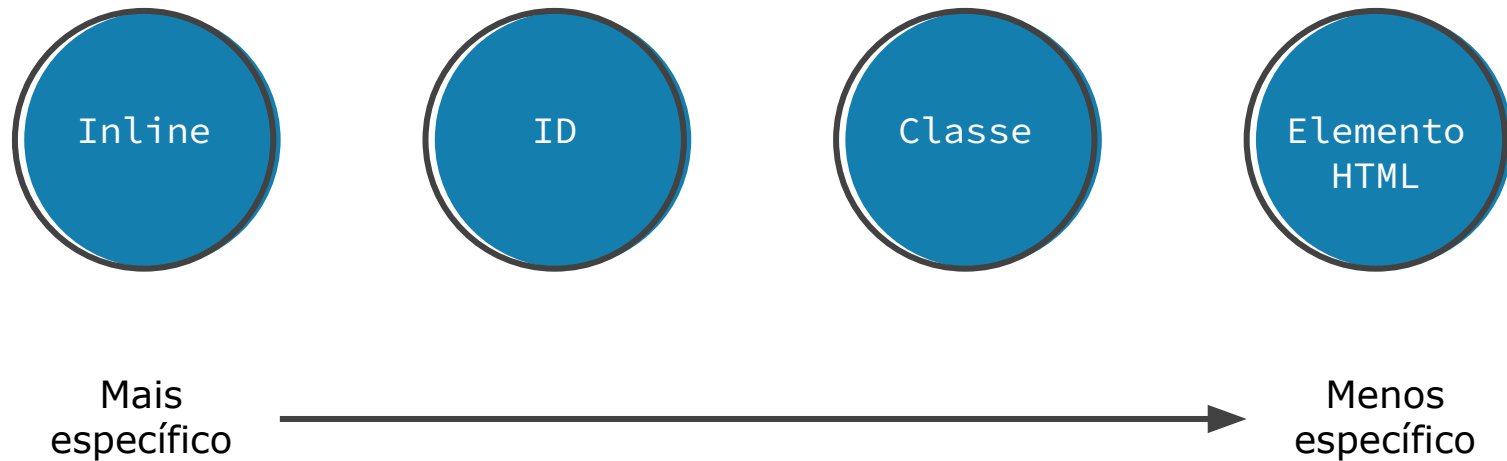


```
<html>
  <head>
    <style>
      h1{
        background-color: black;
      }

      .classe{
        background-color: pink;
      }

      #id{
        background-color: purple;
      }
    </style>
  </head>
  <body>
    <h1 class="classe" id="id" style="background-color:blue;">
      Hello world!
    </h1>
  </body>
</html>
```

Especificidade



Dica para o Desafio: PESQUISA POR FLEXBOX CHEATSHEET

Acesse:

<https://barbaratieko.github.io/tecweb/aulas/02-desafio-css/>

Entrega: **23/02 (via GitHub Classroom)**



Até a próxima aula!

Barbara Agena

barbarata@insper.edu.br