



UNIPÊ
Centro Universitário
de João Pessoa

APLICAÇÕES PARA INTERNET

PROF. DANIEL BRANDÃO

@PROFDANIELBRANDAO

O QUE VIMOS ATÉ
AQUI?



O que vimos até aqui?

- Arquitetura e Protocolos de rede/internet
- Camadas de sistemas Web
- Tecnologias básicas da Web
- Primeiros passos com HTML
- TAGs de texto, link e imagem
- Listas, tabelas e Container
- Formulários e Inputs
- **Estilo e Layout**

O QUE VAMOS
VER HOJE?



O que vamos ver HOJE?

- Grids e Flexbox para Criação de Layouts



The background is a dark blue gradient with a subtle pattern of white dots. Overlaid on the left side is a large, semi-circular degree scale ranging from 150 to 260. Several concentric circles and dashed lines with arrows are scattered across the slide, creating a technical or scientific aesthetic.

MAS PRIMEIRO...

RESULTADO DA AULA PASSADA!

PÁGINA HTML

HTML Linguagem de Marcação de Hipertexto

Tutorial para Iniciantes



HTML (Linguagem de Marcação de HiperTexto) é o bloco de construção mais básico da web. Define o significado e a estrutura do conteúdo da web. Outras tecnologias do HTML geralmente são usadas...

Tabela de TAGs HTML

Tabela de TAGs HTML

Tag HTML	Para que Serve
HTML	define um documentos HTML
HEAD	Define a área de configuração de um documento HTML
BODY	Define a área de conteúdo de um documentos HTML
A	Tag de link (ancoragem) em elementos dentro de um doc HTML
IMG	Define uma imagem dentro de um doc HTML
Div	Elemento que divide e organiza conteúdos dentro de um doc HTML

Top 5 protocolos

1. HTTP - HyperText Transfer Protocol
2. FTP - File Transfer Protocol
3. POP - Post Office Protocol
4. SMTP - Simple Mail Transfer Protocol
5. IMAP - Internet Messaging Access Protocol

- HTTP - HyperText Transfer Protocol
- FTP - File Transfer Protocol
- POP - Post Office Protocol
- SMTP - Simple Mail Transfer Protocol
- IMAP - xxxxxxxx

Código Fonte disponível em:

<https://github.com/danielbrandao/appinternet>

FLEXBOX E GRID

Flexbox e Grid

- Já vimos que utilizar Containers e Estilos nos permite editar um código HTML mais limpo e bem organizado.
- Mas, além disso, o uso de estilos nos permite diagramar melhor uma página web, possibilitando deixar uma página responsiva e pronta para ser aberta em qualquer resolução de tela.

Flexbox e Grid

- O **Flexible Box Model** - modelo de caixa flexível - também conhecido como Flexbox, é um modo mais eficiente para criar layouts, alinhar e distribuir espaços entre itens em um container, mesmo quando as dimensões destes itens são desconhecidas e/ou dinâmicas.

Flexbox e Grid

- Esse modelo é definido como **unidimensional**, o que significa que conseguimos controlá-lo em apenas uma dimensão (ou direção) de cada vez.
- De maneira mais prática: ou controlamos as linhas de um elemento flex, ou controlamos as colunas, os dois não é possível.

Flexbox e Grid

- A área de um documento HTML que faz a utilização do flexbox é chamada de **flex container** (ou container flex).
- Para definir essa seção no documento, basta passar o valor flex ou inline-flex para a propriedade display do CSS.
- O que vai dentro desta área, ou container, são os itens. Eles podem ser desde um texto, até elementos complexos, como um card, por exemplo.

Flexbox e Grid

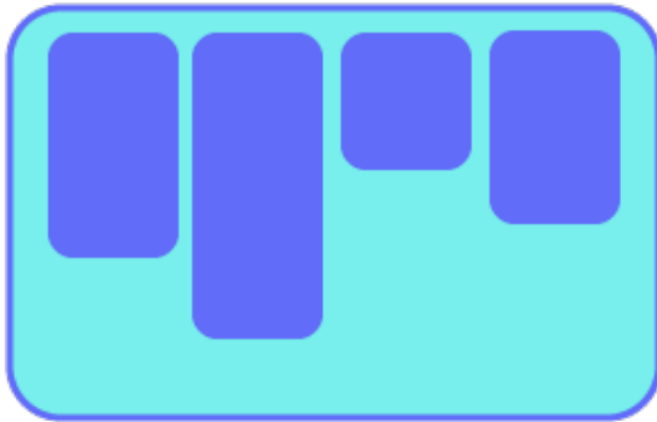
Quando utilizamos o flex, podemos ter duas directions:

- **row**: itens agrupados horizontalmente.
- **column**: itens agrupados verticalmente.
- Uma das propriedades mais poderosas que o flexbox traz para o CSS é o “**align-items**” = alinhamento de itens.

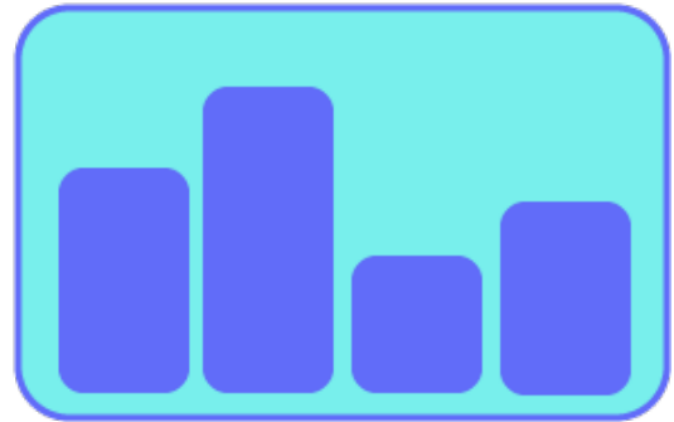
Box-sizing e border-box

- A seguir, veremos 5 maneiras de alinhar os itens do flexbox usando a propriedade **align-items**:

flex-start

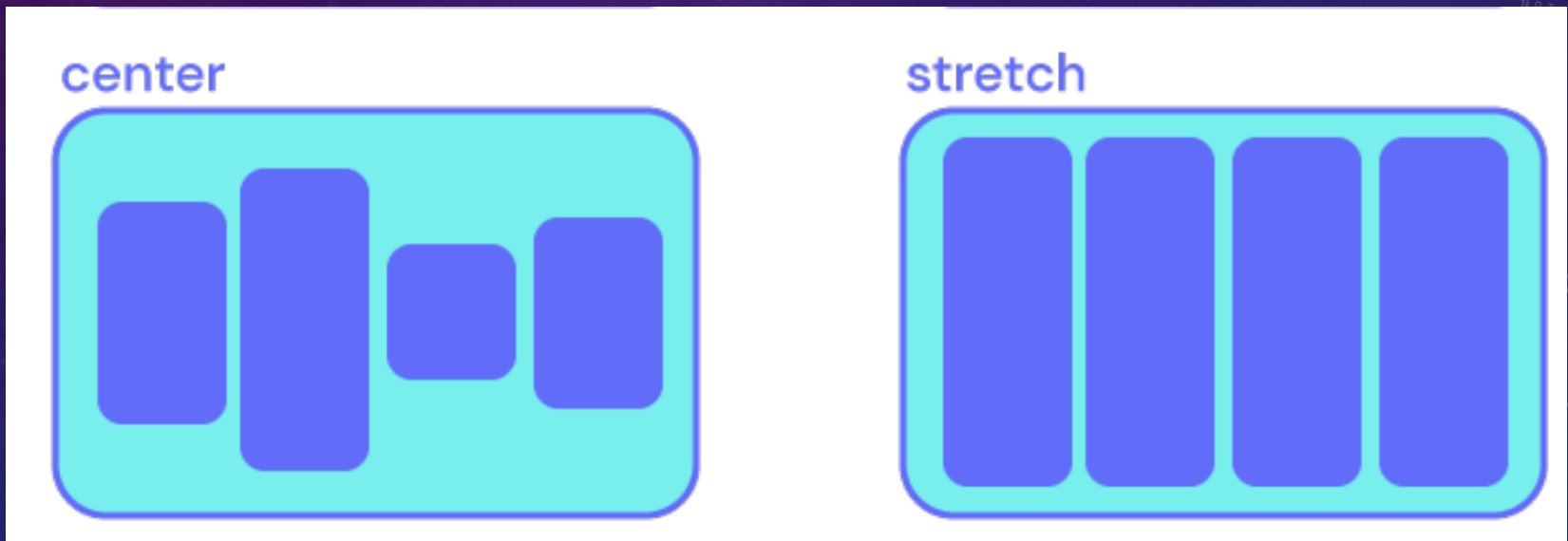


flex-end



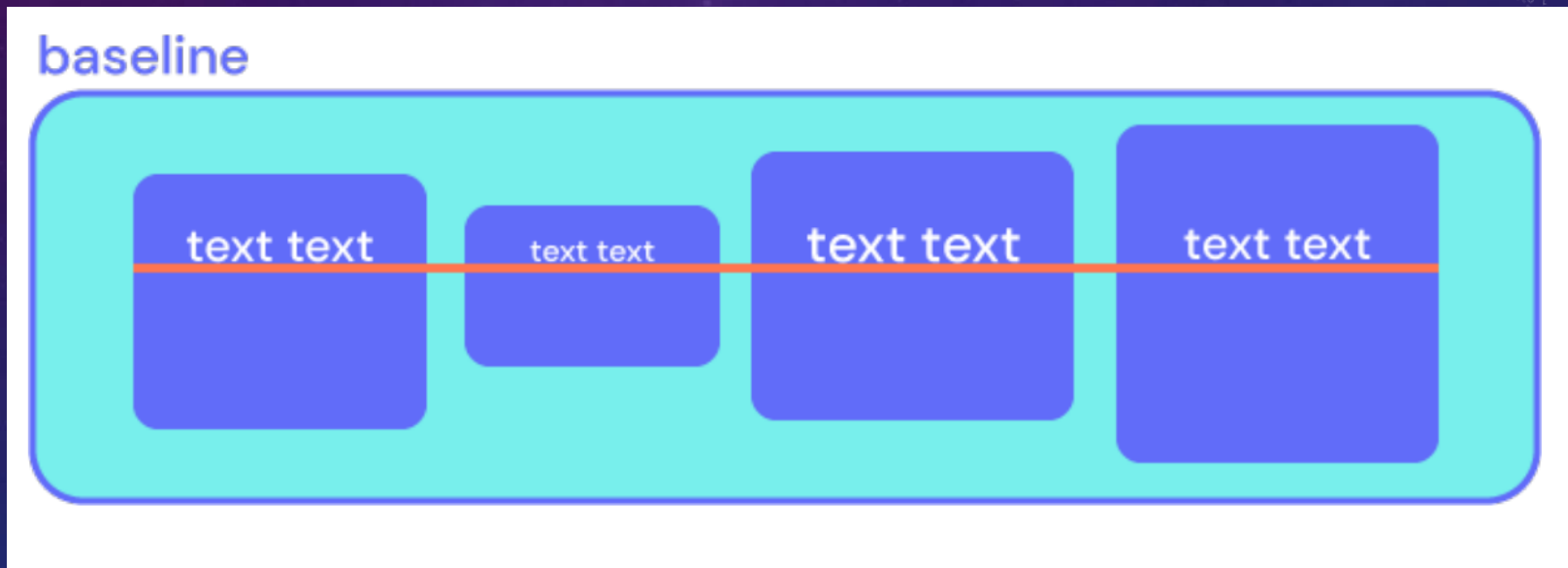
Box-sizing e border-box

- 5 maneiras de alinhar os itens do flexbox usando a propriedade align-items



Box-sizing e border-box

- 5 maneiras de alinhar os itens do flexbox usando a propriedade align-items



Box-sizing e border-box

- 5 maneiras de alinhar **textos** do flexbox usando a propriedade align-items

Flex-start

Alinha os itens ao começo do container.

Flex-end

Alinha os itens ao final do container.

Box-sizing e border-box

- 5 maneiras de alinhar **textos** do flexbox usando a propriedade align-items

Center

Alinha os itens ao centro.

Stretch

É o padrão do align-items. Deixa todos os itens lado a lado e também “estica” os itens, tentando preencher todo o container.

Box-sizing e border-box

- 5 maneiras de alinhar **textos** do flexbox usando a propriedade align-items

Baseline

Alinha o texto dentro dele, de forma que todos fiquem alinhados.

Box-sizing e border-box

“justify-content” = justificar conteúdo

- Essa propriedade alinha horizontalmente em relação ao container. O padrão dessa propriedade é alinhar os itens na horizontal.

IMPORTANTE

- Quando você quer dispor os itens como uma lista, use: **column**.
- Quando você quer dispor os itens de maneira lateral, use o **row**.

Alinhamento Justificado

flex-start



flex-end



center



Alinhamento Justificado

space-between



space-around



space-evenly



Alinhamento Justificado

Flex-start

Alinha os itens no começo do container.

Flex-end

Alinha os itens ao final do container.

Alinhamento Justificado

Space-between

Adiciona um espaço entre os elementos, de forma que eles fiquem o mais separados possível.

Space-around

Adiciona o espaçamento antes do item
após o item e entre os itens.

Space-evenly

Faz com que o espaçamento entre os itens seja igual.

Alinhamento Justificado

Flex-start

Alinha os itens no começo do container.

Flex-end

Alinha os itens ao final do container.

Center

Alinha os itens ao centro.

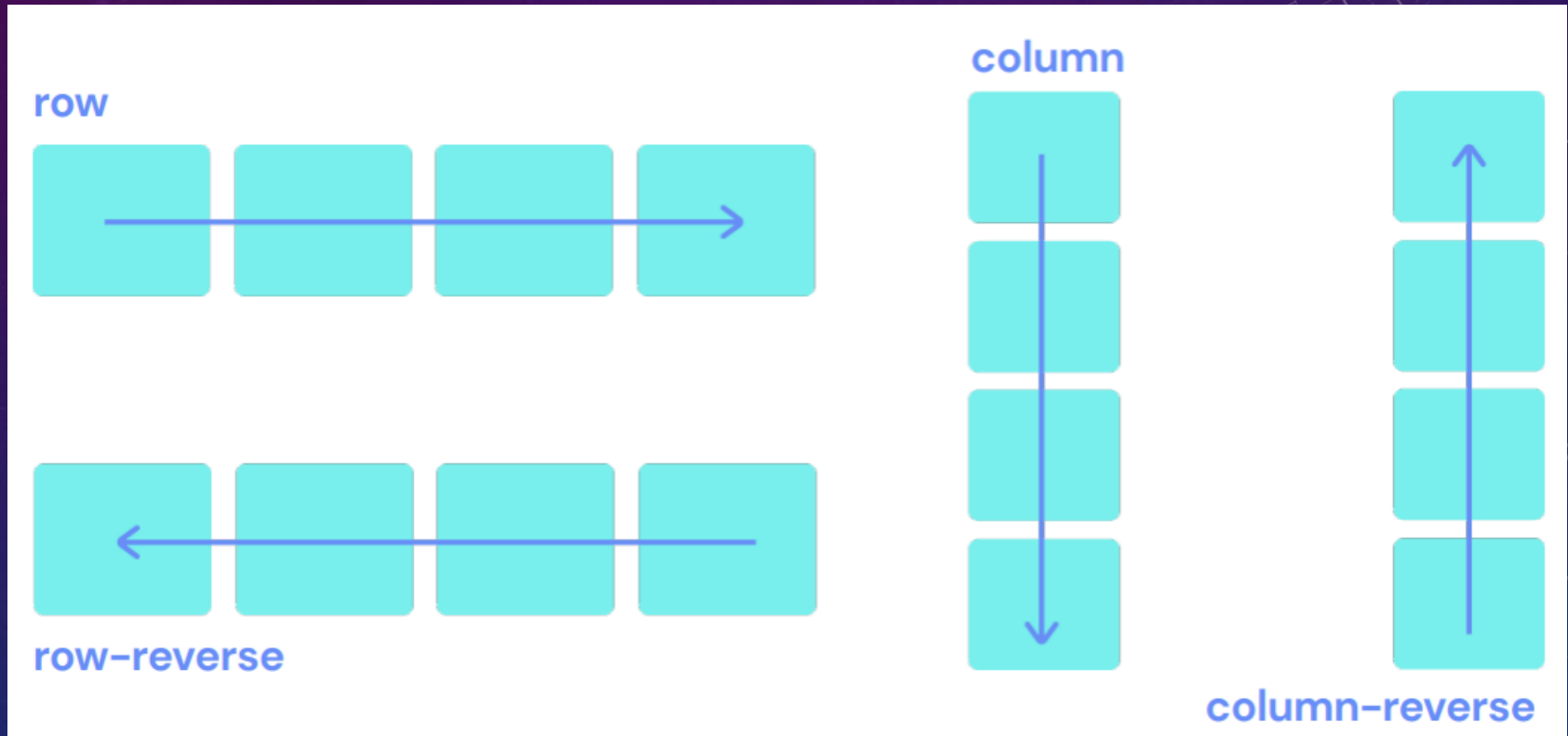
The background is a dark blue gradient with a subtle pattern of white dots. Overlaid on this are several white circular elements: a large scale on the left with markings from 150 to 260, and several smaller concentric circles with arrows indicating clockwise or counter-clockwise rotation. The title 'DIREÇÃO DE FLEXBOX' is centered in a large, white, sans-serif font.

DIREÇÃO DE FLEXBOX

DIREÇÃO DE FLEXBOX

- Podemos Aplicar direção ao flex usando o valor `Flex-direction` (direção flexível)
- A direção pode ser a nível de **Linha (row)** ou **Coluna (column)**
- Row é o padrão. Se você não colocar a propriedade `flexdirection` no seu código, a direção do flexbox vai ser sempre row.

DIREÇÃO DE FLEXBOX



IMPORTANTE

Row-reverse e Column-reverse

- Semelhantes ao “row” e ao “column”, eles reverterem os itens, ou seja, eles ficam ordenados de trás pra frente

DIREÇÃO DE FLEXBOX

Float x Flex

- O **float** tira os elementos do fluxo natural da árvore de elementos do HTML. Isso faz com que seja mais difícil de lidar com o seu posicionamento.
- O **flexbox** possui muitas propriedades que auxiliam e facilitam na hora de ajustar o posicionamento de itens e como eles irão se comportar dentro do *container*.

DESAFIO PRÁTICO!

- Vamos alterar nosso documento anterior, inserindo o atributo **DISPLAY** com valor **FLEX** para as **DIVS** e/ou **SECTIONs** da Tabela e da Lista ordenada.
- Altere os *containers* que contém a Tabela e a Lista, substituindo o *FLOAT LEFT/RIGHT* por *JUSTIFY-CONTENT* de acordo com um dos exemplos apresentados.
- **Exemplo:** `<div style="display: flex;">`

The background is a dark blue gradient with a subtle pattern of white dots. Overlaid on this are several white geometric elements: a large circular scale with degree markings from 150 to 260, and several concentric circles of varying sizes, some with arrows indicating a clockwise direction.

UTILIZANDO GRID

GRID

Display grid = grade de exibição

- Traz facilidade para a organização padronizada de vários elementos.
- Enquanto **Flexbox** é para alinhamento, **Grid** é para layout da página.

FLEXBOX X GRID

- Apesar do flexbox ser capaz de lidar com linhas e colunas, assim como o grid, eles têm propósitos diferentes e complementares.
- **Flexbox**: unidimensional
- **Grid**: bidimensional

FLEXBOX X GRID

O Grid, além de trabalhar com itens unidimensionais, consegue colocar um mesmo item para ocupar espaço em linhas e colunas diferentes.

- Ambos precisam ser aplicados a um container.
- A diferença está que no flexbox a maior parte da definição é feita nos itens do container, enquanto que no grid acontece no container.

FLEXBOX X GRID

- Por padrão, ao colocar **display grid** em um container, os itens ficarão alinhados na vertical, e os espaços serão divididos igualmente entre os itens.
- **Exemplo:** `<div style="display: grid; justify-content: flex-start">`

FLEXBOX X GRID

- A propriedade “**grid-template-columns**”, informa para o container quantas colunas o grid deve ter.
- **Exemplo para inserir 3 colunas:** `<div style=“grid-template-columns: auto auto auto”>`

The background is a dark blue gradient with a subtle pattern of white dots. Overlaid on the left side are several concentric circles and a large circular scale with degree markings from 150 to 260. Some circles have arrows indicating a clockwise direction. The overall aesthetic is technical and modern.

GRID COLUMNS

GRID-COLUMNS

- A propriedade **grid-columns** é uma maneira abreviada de especificar o tamanho e posicionamento de um item dentro de uma grid.
- Ao determinar o começo e o fim de um intervalo numérico, o número que determina o fim do intervalo não é incluído na contagem.

Grid-column-start

Informa em qual coluna o item deverá ser iniciado.

Grid-column-end

Informa em qual coluna o item terminará.

GRID-COLUMNS

- Tudo que se aplica para colunas, também é aplicado aqui para as linhas.
- A única diferença é o nome da propriedade, que ao invés de ser “**grid-template-columns**” é “**gridtemplate-rows**”.
- Para adicionar um espaço entre os itens, podem ser utilizadas as seguintes propriedades:

Column-gap

Espaço entre as colunas.

Row-gap

Espaço entre as linhas.

GRID-COLUMNS

- Para criar layouts com itens que ocupam mais de uma posição no grid, pode ser usado:

Grid-area

Informa como irão funcionar as áreas do grid.

Grid-template-areas

Especifica como os itens devem estar dispostos dentro do grid.

GRID-COLUMNS

- Para criar layouts com itens que ocupam mais de uma posição no grid, pode ser usado:

Grid-area

Informa como irão funcionar as áreas do grid.

Grid-template-areas

Especifica como os itens devem estar dispostos dentro do grid.

DESAFIO MÃO NA MASSA!

- **Usando grid-template-areas, crie uma estrutura para o site contendo as áreas: TOPO, Menu, Seção 1 com 2 colunas, Seção 2 com 3 colunas (lado a lado), Seção 3 abaixo com 4 colunas de conteúdo, rodapé (footer).**
- **OBS: Não precisa enviar a atividade, traga ela na próxima aula para finalizarmos.**

DESAFIO MÃO NA MASSA!

HTML Linguagem de Marcação de Hipertexto

Tutorial para Iniciantes



HTML (*Linguagem de Marcação de HiperTexto*) é o bloco de construção mais básico da web. **Define o significado e a estrutura do conteúdo da web.** Outras tecnologias do HTML geralmente são usadas...

Tabela de TAGs HTML

Tabela de TAGs HTML

Tag HTML	Para que Serve
HTML	define um documentos HTML
HEAD	Define a área de configuração de um documento HTML
BODY	Define a área de conteúdo de um documentos HTML
A	Tag de link (ancoragem) em elementos dentro de um doc HTML
IMG	Define uma imagem dentro de um doc HTML
Div	Elemento que divide e organiza conteúdos dentro de um doc HTML

Top 5 protocolos

1. HTTP - HyperText Transfer Protocol
2. FTP - File Transfer Protocol
3. POP - Post Office Protocol
4. SMTP - Simple Mail Transfer Protocol
5. IMAP - Internet Messaging Access Protocol

- HTTP - HyperText Transfer Protocol
- FTP - File Transfer Protocol
- POP - Post Office Protocol
- SMTP - Simple Mail Transfer Protocol
- IMAP - xxxxxxxx

O que é CSS?

Cascading Style Sheets (CSS) é usado para formatar o layout de uma página da web. Com CSS, você pode controlar a cor, a fonte, o tamanho do texto, o espaçamento entre os elementos, como os elementos são posicionados e dispostos, quais imagens de fundo ou cores de fundo devem ser usadas, exibições diferentes para dispositivos e tamanhos de tela diferentes e muito mais!

O que é HTML?

HTML é uma linguagem de marcação utilizada na construção de páginas na Web. Documentos HTML podem ser interpretados por navegadores. A tecnologia é fruto da junção entre os padrões HyTime e SGML. HyTime é um padrão para a representação estruturada de hipermídia e conteúdo baseado em tempo.

O que é JavaScript?

JavaScript é uma linguagem de programação interpretada estruturada, de script em alto nível com tipagem dinâmica fraca e multiparadigma. Juntamente com HTML e CSS, o JavaScript é uma das três principais tecnologias da World Wide Web.

1

2

3

4

DÚVIDAS?



Referências

- **W3SCHOOLS.** Disponível em: <<http://w3school.com>>.
- MILETTO, Evandro Manara; BERTAGNOLLI, Silvia de Castro. **Desenvolvimento de Software II: Miletto, Evandro Manara; Bertagnolli, Silvia de Castro.** Porto Alegre: Bookman, 2014.
- **W3C.** HTML5 Forms. Disponível em <https://www.w3schools.com/html/html_form_input_type_s.asp>.