GAMA ACADEMY

CSS

Material de Apoio

E-mail: contato@gama.academy

Website: gama.academy

CSS

1ª Edição

Autoria: Kai Pimenta

Editora: Lauren da Silveira Francke

Consultor de Acessibilidade: Eduardo Zangrossi Lino

Consultora de Design: Andressa Gonçalves Cordeiro

Especialista Técnica: Hendy Almeida

Sumário

Clique nos links para ir ao capítulo correspondente:

Abertura

Acessibilidade é para todos!

É com essa frase que começamos uma nova fase na Gama Academy, pensando em um formato mais acessível para nossas apostilas, tornando assim sua jornada conosco muito mais prazerosa e seu aprendizado mais dinâmico.

O que você verá de diferente:

- A começar pelo novo visual dos nossos materiais;
- Trocamos o formato para o vertical, para facilitar a leitura e tornar muito mais fácil e dinâmico para quem usa o celular para estudar;
- O contraste de cores e tamanho das fontes seguem recomendações internacionais de acessibilidade da WCAG;
- Os textos passarão a ser melhor distribuídos para a redução da carga cognitiva necessária para entendê-los, assim passaremos a respeitar ainda mais o tempo de aprendizado de cada um, facilitando para que você possa retornar para o conteúdo que tenha gerado dúvidas com maior facilidade;
- Legendas em todas as imagens do conteúdo para facilitar a identificação;
- Todas as imagens que são necessárias para a devida contextualização sobre o tema, passarão a contar com audiodescrição. É um texto oculto que será lido apenas para pessoas que utilizam tecnologias assistivas, como por exemplo, leitores de tela.

Contamos muito com o feedback de vocês para tornarmos isso um processo de melhoria contínua e nossos materiais atingirem cada vez mais pessoas.

Objetivos de Aprendizagem

A partir do estudo desta apostila, você deverá ser capaz de:

- Compreender o que é CSS e como utilizar;
- Identificar a anatomia de uma elemento CSS;
- Aplicar fontes, cores , textos e backgrounds;
- Compreender um design responsivo.

Esta apostila está dividida em tópicos e em cada um deles você irá encontrar uma pequena introdução sobre o tema e uma recapitulação do que você aprendeu.

Tópico 01

O que é CSS?

Introdução
O que é CSS?

O CSS vem do inglês Cascading Style Sheet, ou seja, folha de estilos em cascata e é usado para estilizar páginas web. O nome de cascata se dá porque para trabalhar com o CSS é necessário seguir uma ordem de tags, classes ou IDs até chegar no componente que você quer estilizar.

O que é CSS?

O navegador lê seu arquivo de cima para baixo, ou seja a ordem que você coloca suas tags, classes e estilos são de extrema importância. Quando você tem duas regras que atingem o mesmo componente, o último a ser declarado vai ser o usado pelo navegador.

Para adicionar uma folha de de estilo no seu projeto é usado a tag <link> no <head> do HTML:

```
<link rel="stylesheet" href="./styles.css">
```

O atributo **rel** é usado para definir qual o relacionamento entre o documento atual e o documento linkado e o **href** indica o caminho para o documento.

Também é possível adicionar o CSS através da tag **<style>** ou adicionando um estilo inline, ou seja, direto na tag:

```
<h1 style="color: red;"> Heading </h1>
```

O estilo inline deve ser usado com cuidado e pontualmente, já que ele tem mais prioridade do que o css exportado de um documento.
Já a tag <style> muitas vezes é desaconselhada a ser usada para evitar poluir o seu html ou deixar ele muito mais pesado do que o necessário.

Resumo do tópico

Neste tópico você aprendeu sobre:

1. O que é CSS e como ele é integrado ao HTML .

Tópico 02

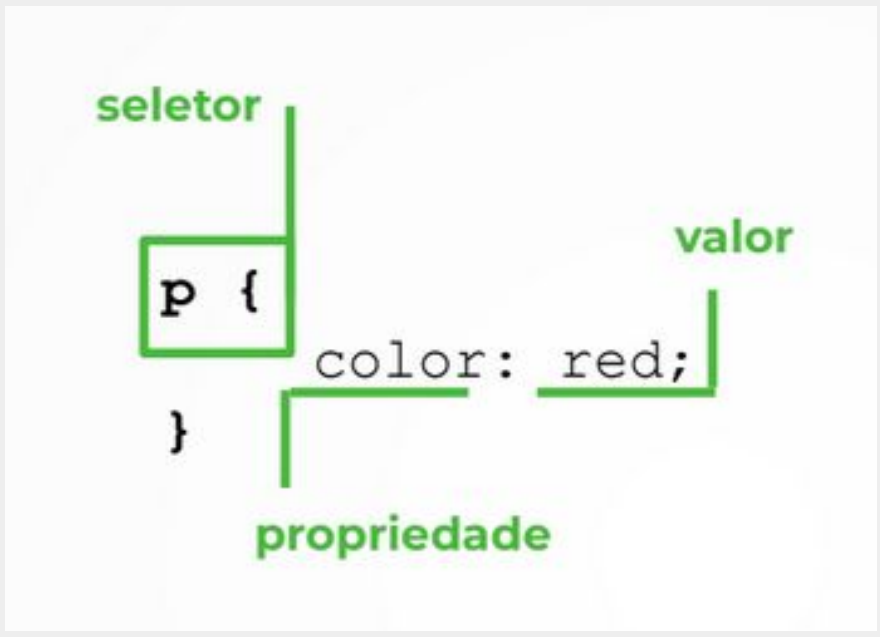
Anatomia de um Elemento CSS

Introdução Anatomia de um Elemento CSS

CSS (Folha de Estilo em Cascata) é o código que você usa para dar estilo à sua página Web. Este código é composto por vários elementos que devem ser escritos seguindo uma regra de estrutura.

Anatomia de um Elemento CSS

Assim como o HTML, o CSS não é realmente uma linguagem de programação. Também não é uma linguagem de marcação — é uma linguagem de folhas de estilos. Isso significa que o CSS permite aplicar estilos seletivamente a elementos em documentos HTML. Por exemplo, para selecionar todos os elementos parágrafo de uma página HTML e tornar o texto dentro deles vermelho, você escreveria este CSS:



Representação da anatomia de um elemento CSS.

Descrição da Imagem

Anatomia de um Elemento CSS

Toda essa estrutura é chamada de conjunto de regras, note os nomes das partes individuais:

- **Seletor (Selector):**

O nome do elemento HTML no começo do conjunto de regras. Ele seleciona o(s) elemento(s) a serem estilizados (nesse caso, elementos <p>). Para dar estilo a um outro elemento, é só mudar o seletor.

- **Propriedades (Property):**

Forma pela qual você estiliza um elemento HTML. (Nesse caso, color é uma propriedade dos elementos <p>.) Em CSS, você escolhe quais propriedades você deseja afetar com sua regra.

- **Valor da propriedade (Property value):**

À direita da propriedade, depois dos dois pontos, nós temos o valor de propriedade, que escolhe uma dentre muitas aparências possíveis para uma determinada propriedade (há muitos valores color(cor) além do vermelho(vermelho)).

- Cada linha de comando deve ser envolvida em chaves ({}).
- Dentro de cada declaração, você deve usar dois pontos (:) para separar a propriedade de seus valores.
- Dentro de cada conjunto de regras, você deve usar um ponto e vírgula (;) para separar cada declaração da próxima.

Resumo do tópico

Neste tópico você aprendeu sobre:

1. Como é a estrutura de um elemento CSS.

Tópico 03

Seletores CSS

Introdução Seletores CSS

Um seletor CSS é a primeira parte de uma regra CSS. É um padrão de elementos e outros termos que informam ao navegador quais elementos HTML devem ser selecionados para que os valores de propriedade CSS dentro da regra sejam aplicados a eles. O elemento ou elementos que são selecionados pelo seletor são referidos como o assunto do seletor.

Há muitos tipos diferentes de seletores, mas nesta apostila mostramos apenas os mais importantes.

Seletores CSS

- **Por Tag:**

Para selecionar uma tag HTML só é necessário colocar o nome da tag, sem ponto, sem #, sem nada. por exemplo:

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

- **Por ID:**

Para selecionar um ID, você deve conferir no seu html o ID usado e colocar:

```
#seu-id {
    color: red;
}
```

- **Por Classe:**

Para selecionar uma classe você deve conferir no seu html a classe e colocar:

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

- **Atenção:**

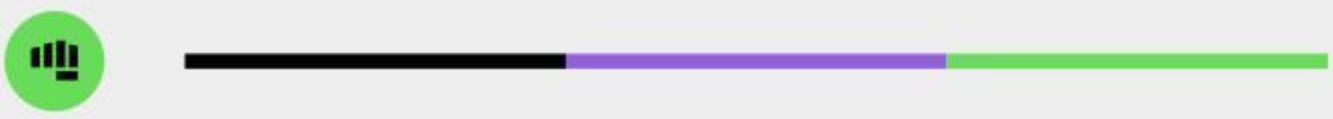
Para selecionar um **id** sempre utilizamos o **#**, para as **classes** o ponto **.** e para **tags** html não usamos **nada** antes.

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

- **Combinando seletores:**

Imagine que nós temos o seguinte HTML:

```
<section class=”minha-classe” >
    <div>
        <p> meu texto </p>
    </div>
        <p> meu texto </p>
</section>
```



Como poderíamos adicionar um estilo apenas para o <p> dentro da ?

Se a gente colocasse no nosso arquivo de css o seguinte:

```
p {
    color: red;
}
```

Todos os <p> da página iriam ficar com esse estilo, que não é o que queremos, por isso podemos combinar seletores. Nesse exemplo poderíamos fazer o seguinte:

```
.minha-classe div p {
    color: red;
}
```

Existem muitas outras formas de concatenar seletores para especificarmos exatamente qual elementos queremos adicionar um estilo.

Caso tenha interesse em se aprofundar ainda mais no assunto, sugiro dar uma olhada nesta explicação no site MDN Web Docs.

Representação dos tipos de seletores.

Descrição da Imagem

Resumo do tópico

Neste tópico você aprendeu sobre:

1. O que é um seletor CSS e como utilizá-lo.

Tópico 04

Fontes e textos

Introdução Fontes e textos

A tipografia na Web é uma parte importante da experiência do usuário. Você deseja garantir que as fontes sigam as diretrizes da marca corporativa e que o conteúdo seja exibido conforme o esperado em vários dispositivos.

Dezenas de unidades CSS diferentes estão disponíveis para definir o tamanho do texto. Várias propriedades CSS estão disponíveis para controlar o tamanho da fonte, o espaçamento, a altura da linha e outros recursos tipográficos.

Fontes e textos

Definir as fontes e a formatação de textos em um projeto é uma das principais características que dão cara ao nosso website.

Algumas fontes já são nativas de navegadores, como por exemplo a Arial e a Verdana, mas muitas vezes queremos usar outras fontes mais diferenciadas e nesses casos sugiro procurar no Google Fonts.

Sobre a formatação do texto algumas propriedades são mais comuns:

- **font-family:** onde você declara a fonte a ser usada.
- **font-weight:** qual o peso dessa fonte? Aqui colocarmos se é bold, light, normal, etc...
- **font-size:** qual o tamanho da fonte.

Também podemos estilizar nosso texto de outras formas, como por exemplo:

- **text-transform:** podemos colocar em caixa alta, caixa baixa...
- **letter-spacing:** dar espaçamento entre as letras
- **line-height:** altura de cada linha de texto

Resumo do tópico

Neste tópico você aprendeu sobre:

1. A importância das fontes em seu website e como utilizar algumas propriedades para estilizar seu texto com CSS.

Tópico 05

Unidades

Introdução Unidades

Assim como na física, o CSS também utiliza unidades de medida como o píxel, centímetro, metro, etc...
Essas unidades que são comuns no nosso dia a dia chamamos de Medidas Absolutas.
Mas existem outras unidades de medida que são um pouco diferente do que estamos acostumados, como o **em** e o **rem** que são utilizadas em layouts digitais e essas chamamos de Medidas Relativas.

Unidades

Você provavelmente já conhece o pixel, essa medida é muito comum quando falamos sobre a web mesmo quando não desenvolvemos, já no css a gente usa “px” quando estamos usando essa unidade, por exemplo:

```
h1 {
    font-size: 12px;
}
```

Também é comum usar o vh (viewport height) e o vw (viewport width), ajudam principalmente em layouts responsivos, uma vez que essas unidades calculam o tamanho do dispositivo e a partir daí coloca uma altura ou largura relativa, evitando que o seu layout quebre em determinados tamanhos.

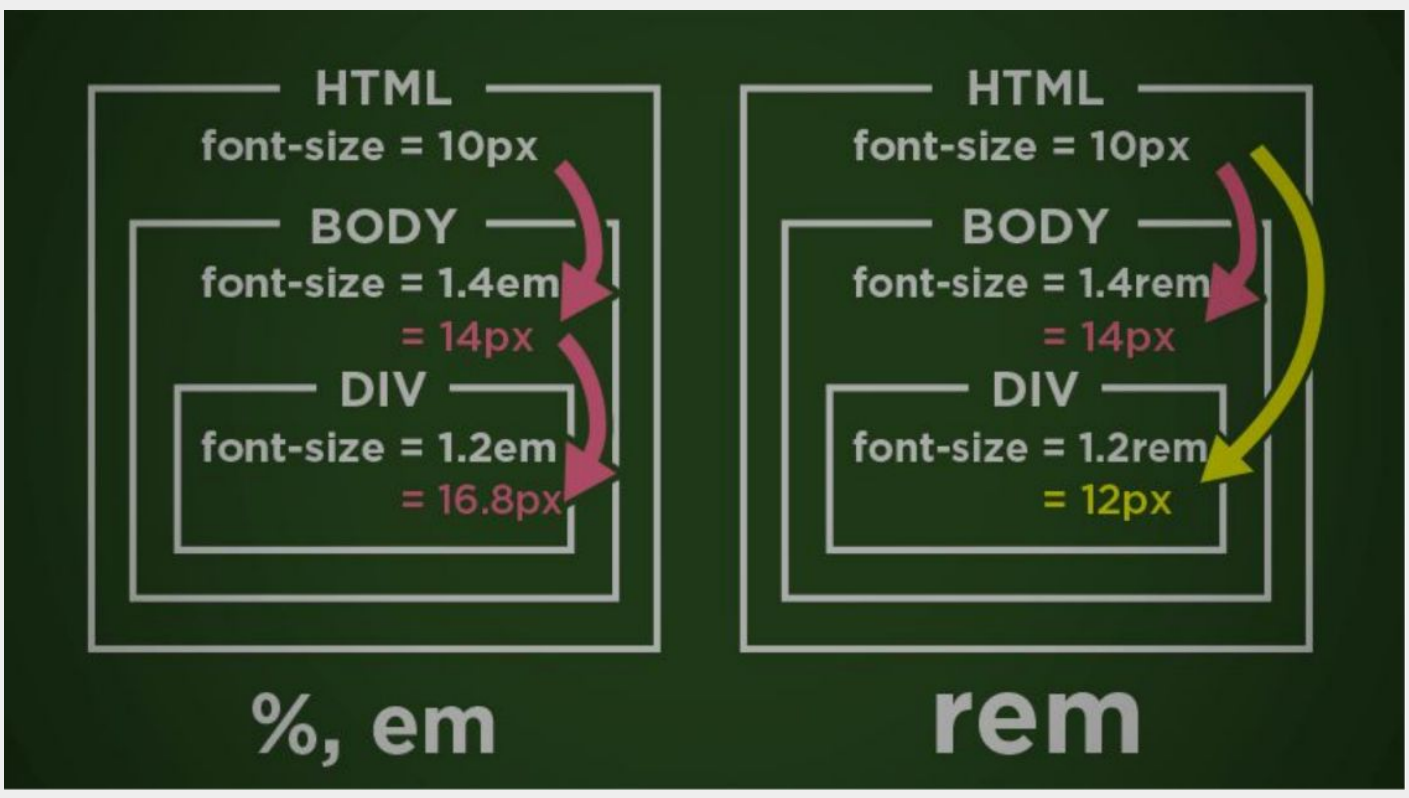
01 - EM:

EM é mais uma unidade relativa, isso significa que por exemplo, se você coloca que todo o <body> tem uma font-size: 14px; e depois colocar que uma que está dentro desse body tem uma font-size: 1.5em, ela vai ter 21px uma vez que 1.5(em) x 14(px) = 21

Atenção:
Quando você usa a unidade EM tome cuidado pois o CSS é um estilo em cascata então toda propriedade com tamanho relativo é relativo ao elemento pai dele, que é relativo ao pai e assim por diante.

2 - REM:

Para solucionar o problema que o em acarreta, também temos a unidade chamada de rem, que funciona muito similar ao em mas o R vem de root, ou seja, é relativo a raíz do projeto, aquele primeiro font-size que colocamos no <body> ou <html>.



Representação da comparação das unidades EM e REM para definição do tamanho de fontes.

Descrição da Imagem

Resumo do tópico

Neste tópico você aprendeu sobre:

1. As principais unidades do CSS e como utilizá-las.

Tópico 06

Cor e background

Introdução Cor e background

A propriedade CSS **background** tem como principal propriedade a manipulação do plano de fundo de uma página ou dos elementos nela declarados e a propriedade **color** do CSS define o valor da cor de um elemento.

Cor e background

Existem duas formas de adicionar cor a um site, dependendo do seu objetivo:

Para adicionar cor a uma fonte, usamos o atributo color, já para adicionar cor a um elemento inteiro, normalmente usamos o background-color.

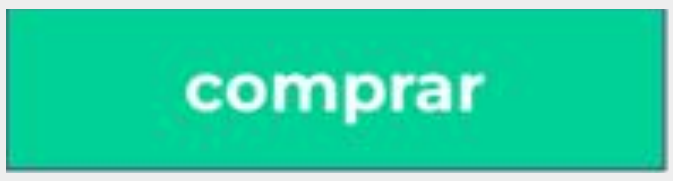
Também é possível adicionar uma imagem a um background, nesse caso usamos o atributo background-image.

Para saber mais sobre o uso do atributo background, você pode acessar esta explicação.

Por exemplo, se vamos fazer um botão podemos usar o seguinte:

```
button {
    color: #fff;
    background-color: green;
}
```

o resultado seria:



Resumo do tópico

Neste tópico você aprendeu sobre:

1. Como adicionar cor a um elemento do site.

Tópico 07

Display inline, block e inline-block

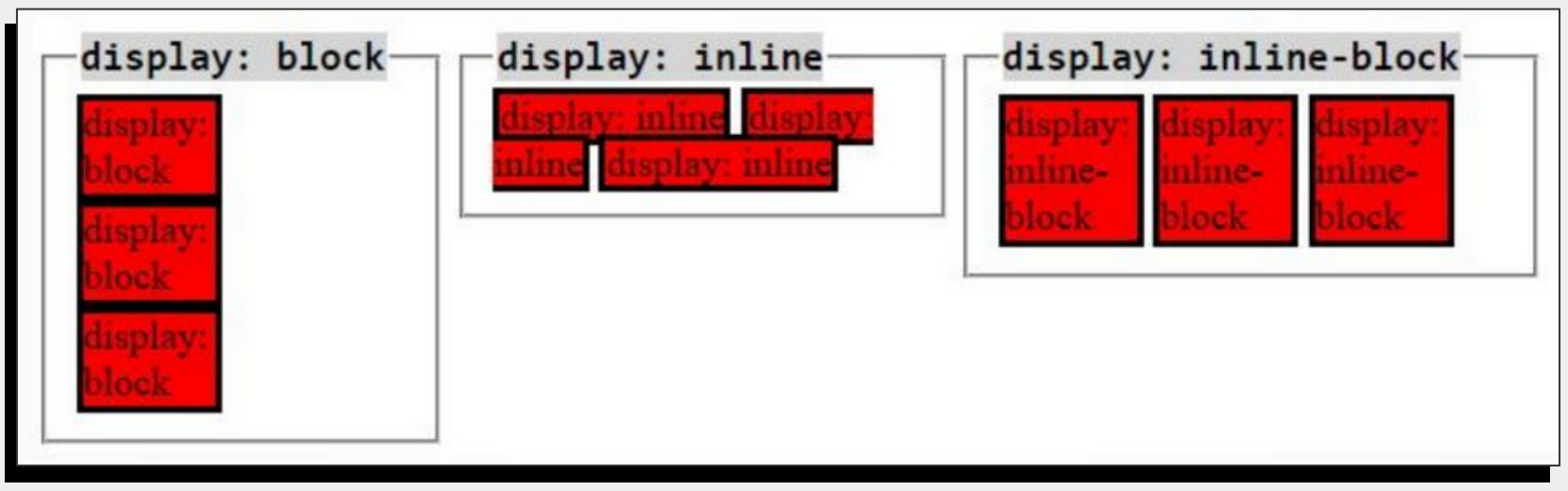
Introdução O que são elementos block e inline?

- block: O elemento block sempre começa em uma nova linha e se expande para esquerda e direita o tanto quanto for possível.

- inline: é o elemento de linha padrão. Um elemento de linha pode preencher algum texto dentro de um parágrafo sem quebrar o fluxo daquele parágrafo.

- inline-block: exibe um elemento HTML como um contêiner de bloco de nível inline.

O que são elementos block e inline?



Representação do comportamento da propriedade display com os valores: block, inline e inline-block.

Descrição da Imagem

A propriedade display especifica a exibição de um elemento. É muito importante entender esse funcionamento.

As tags HTML normalmente já vem com display específicos, por exemplo é um display: inline e <p> é um display: block, mas você pode alterar isso de acordo com seu projeto.

inline

Exibe um elemento na mesma linha, um atrás do outro.

Elementos inline só tomam o espaço necessário do conteúdo e colocar propriedades de altura e largura não vão surtir efeito.

block

Enquanto elementos inline são exibidos na mesma linha, elementos em bloco sempre começam uma nova linha (ou um novo bloco), ocupando toda a largura do elemento pai.

inline-block

Exibe um elemento como um display inline mas possui a característica de um elemento em bloco, ou seja, você pode adicionar uma altura e uma largura a elementos inline-block.

Resumo do tópico

Neste tópico você aprendeu sobre:

1. Como ficam os visuais dos elementos block e inline.

Tópico 08

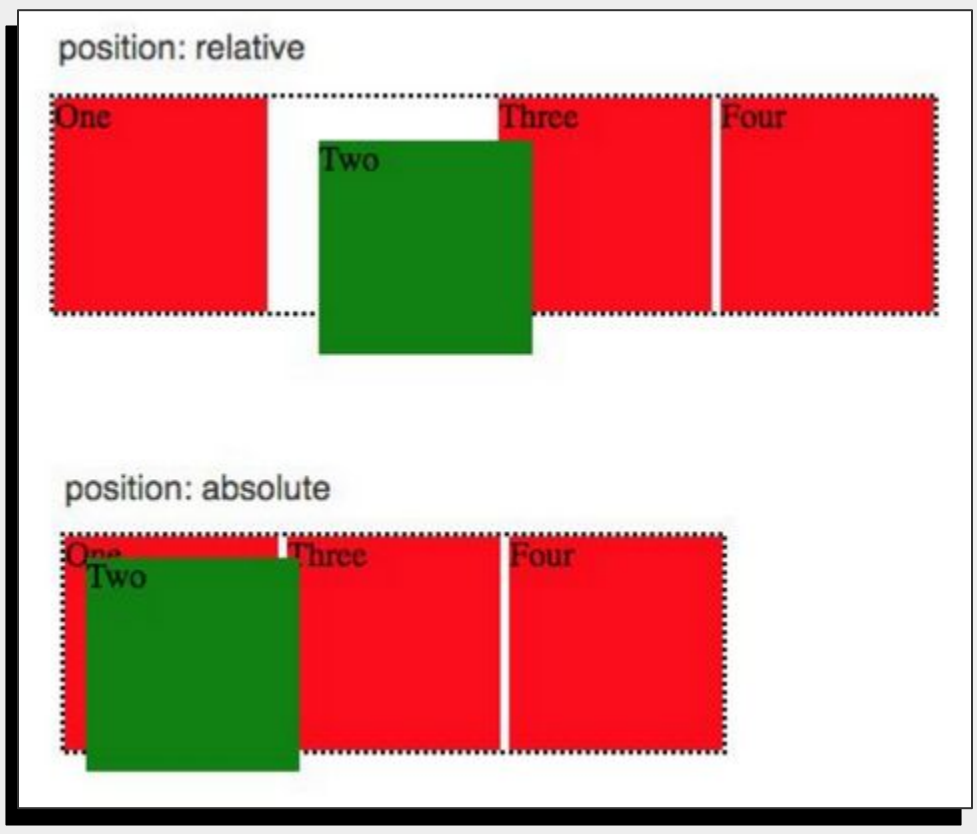
Display e position

Introdução Diferença entre display e position

Enquanto a propriedade display especifica o comportamento de exibição dos elementos (inline, block, inline-block), como visto anteriormente, a propriedade de position especifica a posição do elemento dentro do documento, podendo ser absolute, relative, stick, static ou fixed.

Diferença entre display e position

As propriedades mais comuns de serem usadas são o absolute e relative, no absolute o elemento fica por cima de todos do documento e no relative ele é relativo ao elemento pai.



Representação do comportamento da propriedade position com os valores relative e absolute.

Descrição da Imagem

Para mais informações sobre a propriedade position, acesse esta explicação.

Resumo do tópico

Neste tópico você aprendeu sobre:

1. As diferenças entre display e position.

Tópico 09

Espaçamento

Introdução Espaçamento

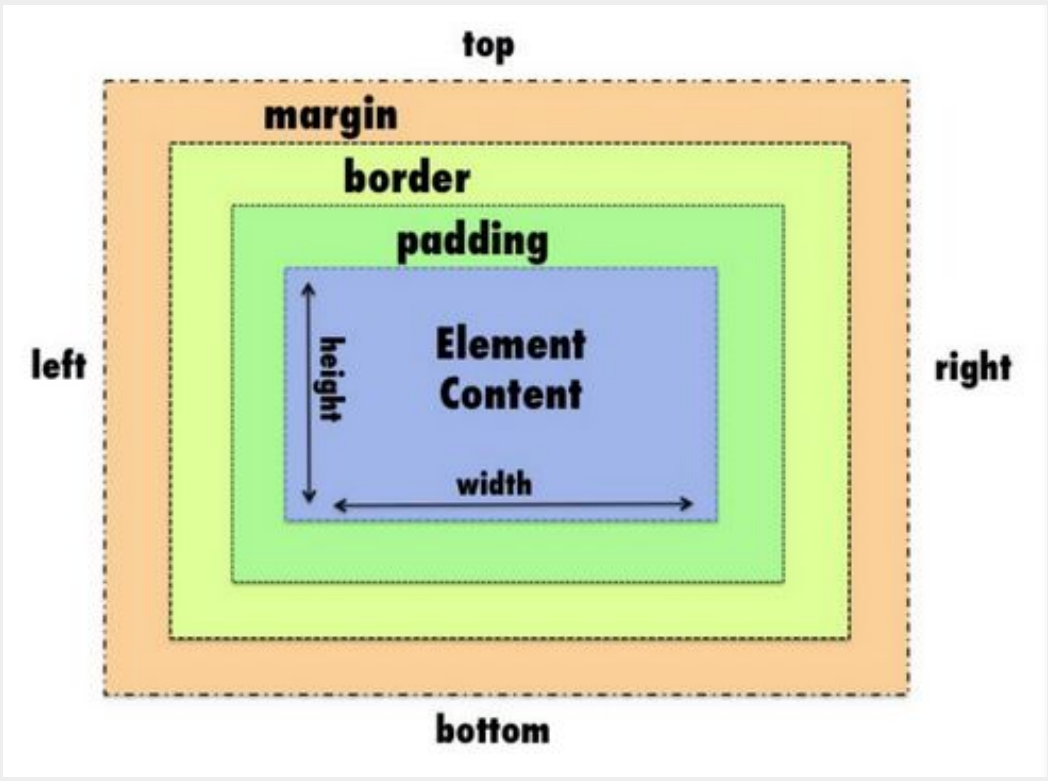
Como esperado, o layout CSS é baseado principalmente no modelo de caixas. Cada um dos blocos que ocupam espaço na sua página tem propriedades como essas:

- padding, o espaço ao redor do conteúdo (ex.: ao redor do texto de um parágrafo).
- border, a linha sólida do lado de fora do padding.
- margin, o espaço externo a um elemento.

Espaçamento

As formas mais comum de você adicionar espaçamento aos seus elementos é através do uso de margens e paddings.

Apesar de funcionarem de uma forma muito semelhante, a principal forma de diferenciar essas duas propriedades é que o padding controla o espaçamento dentro do elemento e a margin controla o espaçamento ao redor do elemento.



Representação de margin, border e padding em um elemento.

Descrição da Imagem

Pontos a serem lembrados

Padding

O padding não vai funcionar em elementos inline, é provável que você precise adicionar um display: inline-block para resolver esse problema.

Quando usar

Use margin para adicionar distância entre elementos.

Use padding para aumentar o tamanho de elementos ou dar uma distância entre o conteúdo e a borda do elemento pai.

Margin

Margin tem a característica de overlap. Ou seja, se vc tem dois componentes um ao lado do outro ou um em cima do outro, ambos com margin, as margens vão colapsar e a maior vai ser usada.

Usando padding os dois paddings seriam usados porque padding é considerado parte do elemento, enquanto margin está do lado de fora do elemento.

Resumo do tópico

Neste tópico você aprendeu sobre:

1. O que são espaçamentos e quando usar padding e margin.

Tópico 10

Reset CSS

Introdução Reset CSS

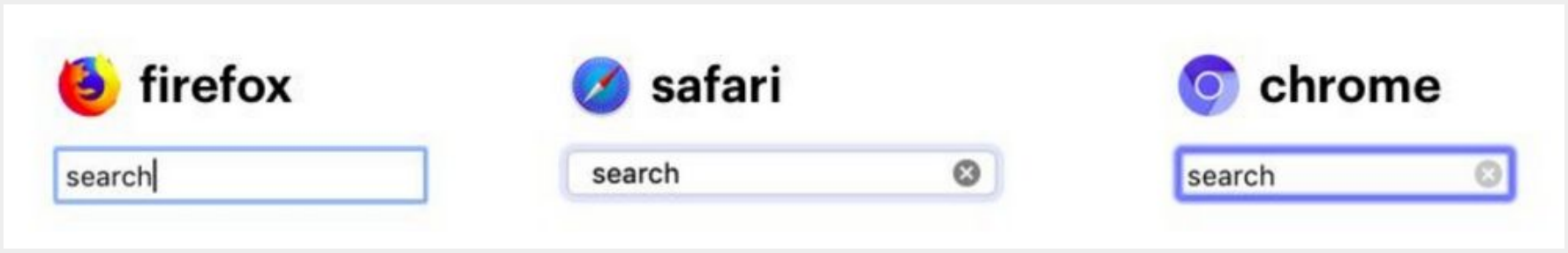
O reset chegou para padronizar a exibição das páginas em diversos tipos de navegadores, pois antes dele os designers tinham muita dor de cabeça para que a página fosse exibida igualmente em todos os navegadores.

O reset sobrepõe uma folha de estilo padrão nos navegadores para remover toda formatação “original” aplicada pelo browser.

Reset CSS

Se você reparar quando inspecionamos um elemento HTML, por padrão já vem com alguns estilos predefinidos que não necessariamente queremos usar em nosso projeto.

Claro que dá pra fazer um site sem resetar o css mas quando não fazemos isso, o navegador vai adicionar suas propriedades padrão e como cada navegador tem suas próprias preferências, seu site pode tem grande chance de ficar diferente entre cada navegador.



Representação dos diferentes estilos para um mesmo input em diferentes browsers.

Descrição da Imagem

Mas qual a melhor forma de resolver isso?

O Reset CSS do Eric Meyer é um dos mais famosos, onde ele zera todas as margins, paddings, remove estilo de listas, outlines... É bem completo!

Você pode copiar esse reset e criar um novo arquivo chamado reset.css e adicionar no seu html, lembre-se de adicioná-lo antes do seu estilo, para que você consiga sobrescrever o reset caso necessário.

Você também pode optar por fazer uma versão mais simples, colocando apenas os pontos que vão ser usados no seu projeto.

Resumo do tópico

Neste tópico você aprendeu sobre:

1. O que é e como utilizar o reset CSS.

Tópico 11

Layout responsivo

Introdução Fazendo um layout responsivo

Segundo o suporte do Google, um design responsivo usa um layout flexível que se adapta a diferentes orientações ou tamanhos de janelas de visualização. Dessa forma, você não precisa criar vários layouts.

Seu documento precisa ter, no mínimo, dimensões de página responsivas. Também é possível criar um layout fluido usando as media querys.

Fazendo um layout responsivo

No desenvolvimento web falamos muito sobre layout responsivo, isso porque de alguns anos pra cá temos visto muito mais necessidade em pensar em como nossas páginas vão ficar em telas maiores ou menores.

Existem várias formas de você desenvolver já pensando na responsividade: você pode tanto usar a técnica do mobile-first, ou seja, desenvolver primeiro para telas de celular e depois escalar para telas maiores ou você pode começar desenvolvendo para telas grandes utilizando algumas propriedades que vão facilitar essa adaptação para celular.

Em ambos os casos é bem provável que você precise usar as media querys.

Uma media query é um pedaço de código que só vai ser renderizado a partir de um tamanho especificado ou até um tamanho especificado.

Você pode usar quantas **media querys** você precisar, por exemplo, no exemplo a seguir temos um media query para a largura máxima de 768px e outro para larguras a partir (min-width) 769px.

Algumas outras dicas que você pode utilizar para evitar que o site fique quebrado e ter a necessidade de ajustar com media querys é usar unidades relativas para tamanho de containers, imagens, textos e etc, como vh e vw, além de usar flexbox e grid.

```
/* Exemple */
/* Mobile and Tablet */

@media (max-width: 768px) {
    html {
        font-size: 16px + 2 * ((100vw - 360px) / 768px);
    }
}

/* Laptop and Desktops screens */
@media (min-width: 769px) {
    html {
        font-size: 14px + 10 * ((100vw - 769px) / 2048px);
    }
}

/* Excessively large screens */
    html {
        font-size: 36px;
    }
```

Resumo do tópico

Neste tópico você aprendeu sobre:

1. O que é um design responsivo e como utilizar as media querys.

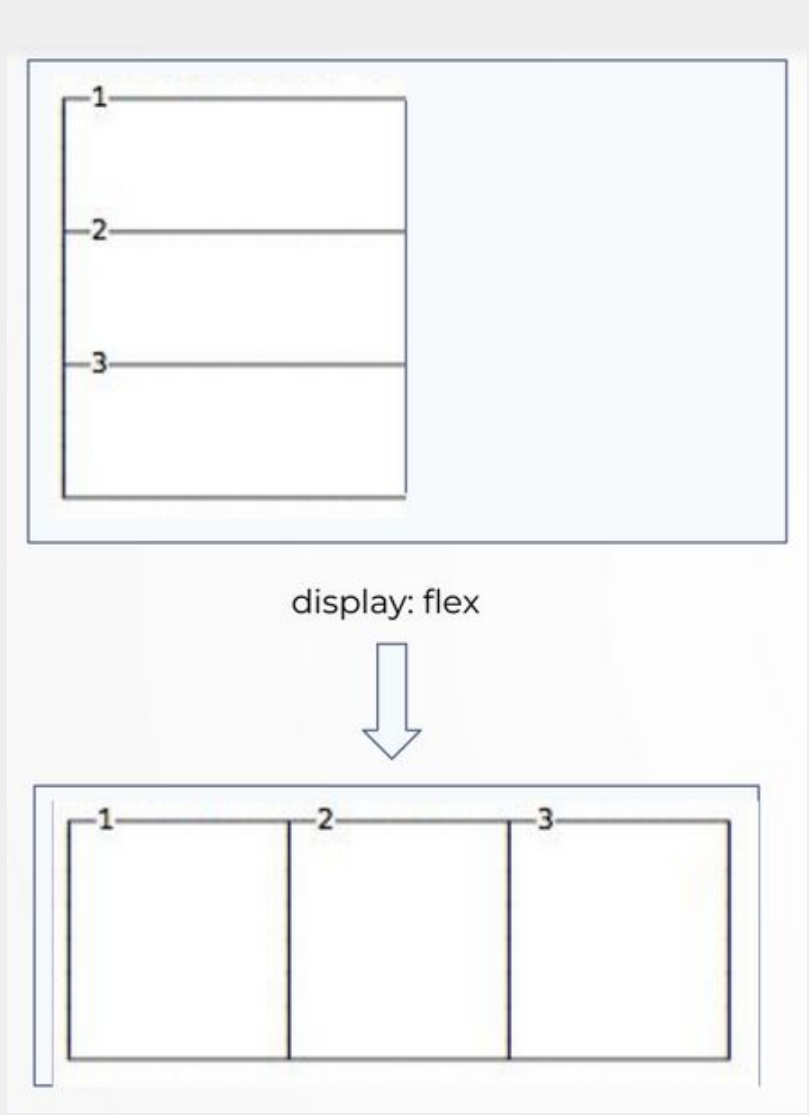
Tópico 12

FlexBox

Introdução FlexBox

O FlexBox é um conceito do CSS3 que visa o layout flexível permitindo que os elementos responsivos dentro de um contêiner sejam organizados automaticamente, dependendo do tamanho da tela.

FlexBox



Representação do uso do display: flex.

Descrição da Imagem

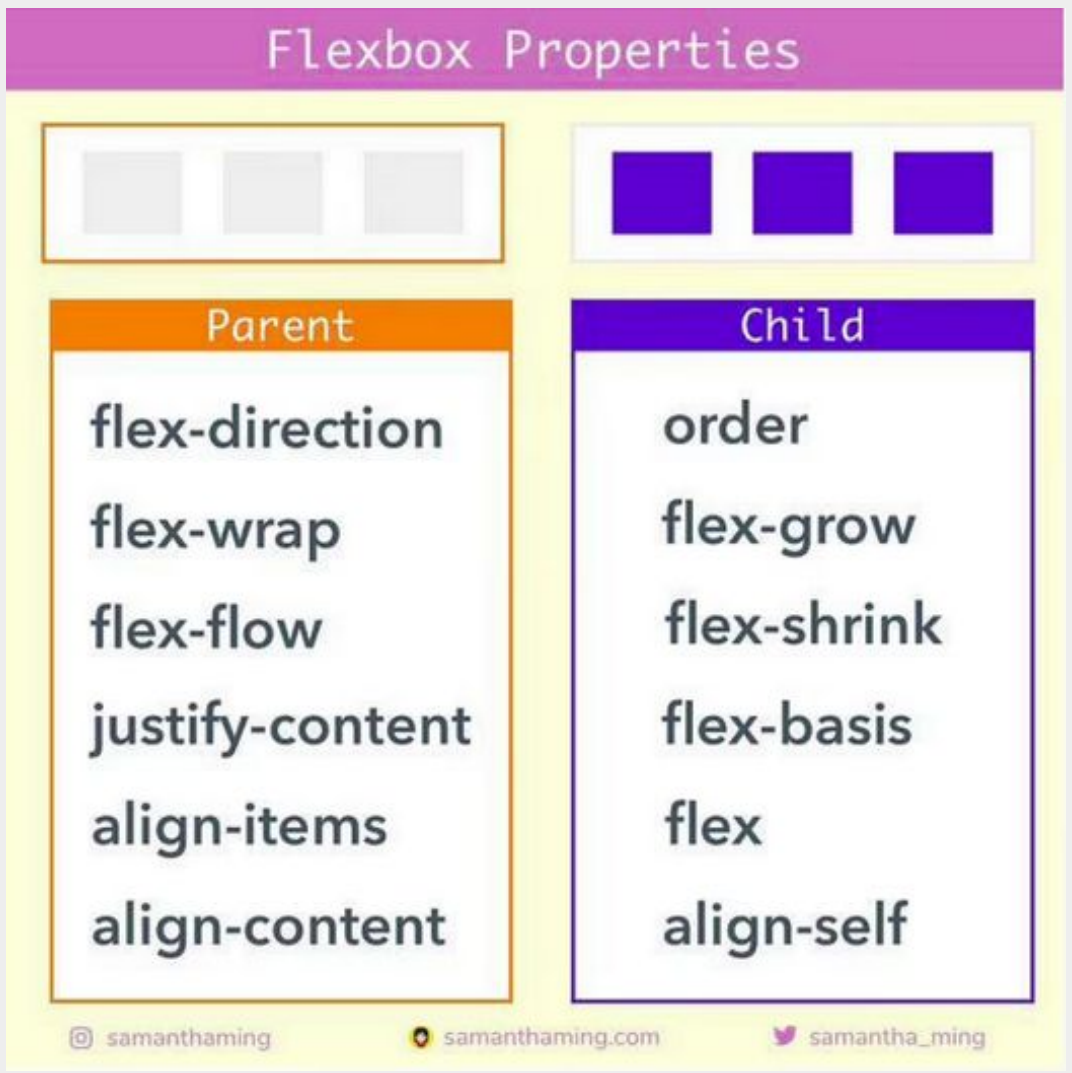
A principal ideia por trás do Flexbox é dar ao container a capacidade de alterar a largura e altura de seus itens para preencher melhor o espaço dele, isso significa que, independente das suas dimensões o conteúdo manterá um layout flexível dentro do seu elemento pai.

Antes da criação do flexbox, ajustes simples como fazer com que todos os elementos tenham o mesmo tamanho dentro de um container ou até mesmo centralizar um elemento ao centro vertical e horizontalmente, era muito complicado ou até mesmo impossível.

Para entender melhor sobre o funcionamento do flexbox, sugiro acessar esta documentação na MDN Web Docs.

Algumas coisas que devem ser levadas em consideração no flexbox é quais propriedades podemos usar no container pai e quais utilizamos nos itens dentro do container.

No container podemos alinhar tanto a direção em coluna quanto em linha usando o flex-direction. Por definição, sempre que usamos o display: flex, o conteúdo fica com o flex-direction: row, ou seja em linha, caso você queira diferente, deve mudar no seu css.



Representação das propriedades do Flexbox.

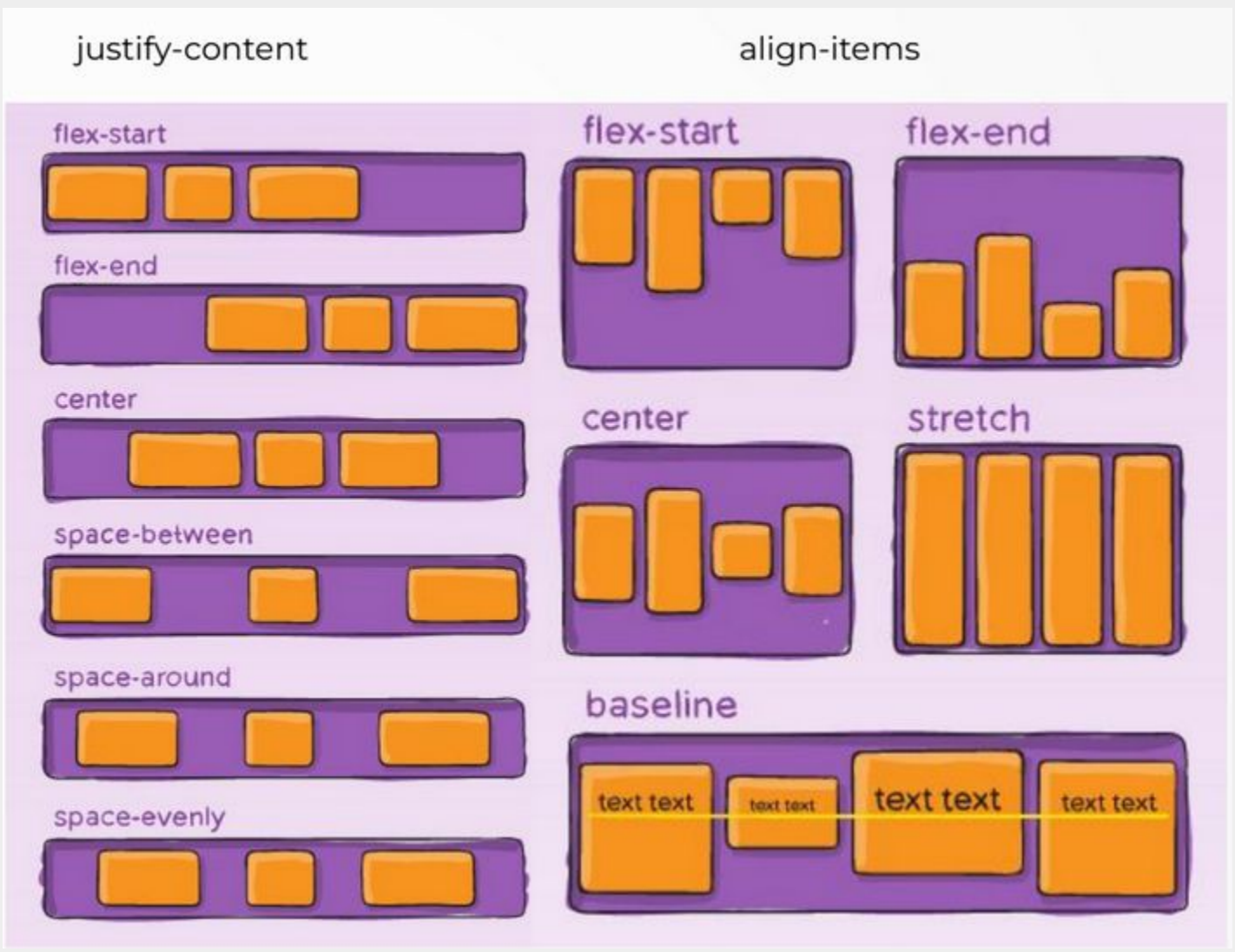
Descrição da Imagem

Uma das propriedades do flexbox mais usadas é a justify-content, que nos ajuda a alinhar o conteúdo dentro do container, no início, no fim, centralizado e com espaço entre ou ao redor. Isso vai depender claro do tamanho do seu container.

Também é muito usado o align-items, isso porque o conteúdo de um container flex é por definição align-items: stretch, por isso que os items se esticam para ocupar toda a altura e largura do container, podemos mudar isso para centralizado e alinhado acima ou abaixo.

Quer aprender sobre flexbox jogando?

Conheça o Flexbox Froggy.



Representação dos valores das propriedades justify-content e align-items.

Descrição da Imagem

Resumo do tópico

Neste tópico você aprendeu sobre:

1. O que é o FleXBox como ele é utilizado para facilitar o layout responsivo.

GAMA ACADEMY