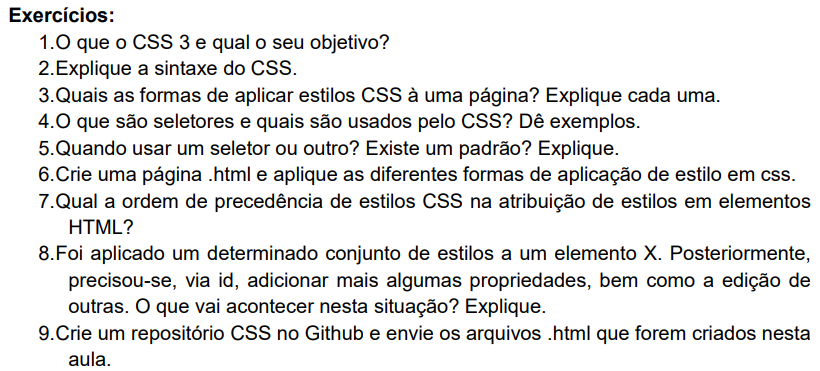
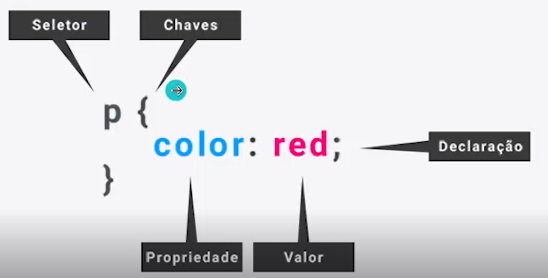
Atividade – 01



Q1) A abreviação CSS vem de Cascading Style Sheets (Folhas de Estilo em Cascata, em português). Trata-se de uma linguagem de estilo usada para descrever a apresentação de um documento escrito em HTML ou em XML. O CSS descreve como elementos são mostrados na tela, no papel, na fala ou em outras mídias. É uma das principais linguagens da open web e atualmente está na versão CSS3, que está se encaminhando para ser uma padronização. Fazendo uma comparação o HTML é o esqueleto, o Javascript é o sistema nervoso e o CSS é a aparência. Cascata está relacionada com a forma que o CSS lida com conflitos, a última estilização é a que prevalece, isto é, no caso de duas estilizações aplicadas ao mesmo elemento, será aplicada a última.

Q2)

Modelo



Por exemplo, considere :p { color:red;}. Nesse caso, p é o seletor, isto é nome da classe, id, tagname, que será alvo e receberá uma estilização; abre-se e fecha “chaves” e dentro fazemos uma ou mais declarações, terminando com “;”. Uma declaração deve conter o ome da propriedade e o valor que ela vai receber. No exemplo comentando, a propriedade é “color” e o valor é “red”. Aplicando-se este estilo, espera-se que a cor do texto fique vermelha.

Q3)

Há três formas: a)inline, b)internal e c)external. Na “inline”, utiliza-se o atributo style dentro da tag de abertura do elemento que desejamos estilizar. Na linha de código “<div style="background-color: blue;"></div>”, o atributo style define o background da “div” como azul. Na internal, é necessário empregar a tag de abertura e fechamento <style>. Essa tag também fica dentro do arquivo html, normalmente é colocado após o <head> antes de começar o <body>. Dentro da tag, utiliza-se a mesma sintaxe comentada na questão 2. Por fim, na forma “external”, é criado um arquivo com extensão .css. Esse arquivo conterá toda a estilização e é necessário estabelecer um vínculo entre o arquivo html e o css. Para isso, dentro da tag <head> do html, devemos adicionar uma tag <link> no padrão: <link rel="stylesheet" href="./caminho/style.css"> sendo importante informar no atributo href o caminho até o arquivo cm extensão .css. De forma geral, utiliza-se a forma external, pois permite uma melhor organização e como futuros desenvolvedores estaremos trabalhando com diversas pessoas e diversos arquivos. Por fim, vale ressaltar que a forma inline tem prioridade sobre as demais, ou seja, qualquer estilização feita “inline” prevalece.

Q4) Os seletores definem a que elemento será aplicado um conjunto de estilizações. São exemplos de seletores: Seletor por tag, Seletor por class, Seletor por ID, Seletor universal (\*), Seletor por atributo. Esses são os seletores básicos. Define-se o seletor e abre e fecha “chaves”. Dentro a gente coloca a estilização através de uma combinação propriedade e valor.

A seguir descrevo os principais usos.

**Seletor por tag**

Este seletor básico escolhe todos os elementos que correspondem ao nome fornecido. Sintaxe: nome-da-tag Exemplo: input corresponderá a qualquer elemento <input>.

**Seletor por classe**

Este seletor básico escolhe elementos baseados no valor de seu atributo classe. Sintaxe: .nome-da-classe Exemplo: .index irá corresponder a qualquer elemento que tenha o índice de classe (provavelmente definido por um atributo class="index", ou similar).

**Seletor por ID**

Este seletor básico escolhe nós baseados no valor do atributo id. Deve existir apenas um elemento com o mesmo ID no mesmo documento. Sintaxe: #nome-do-id Exemplo: #toc irá corresponder ao elemento que possuir o id=toc (definido por um atributo id="toc", ou similar).

**Seletores universais**

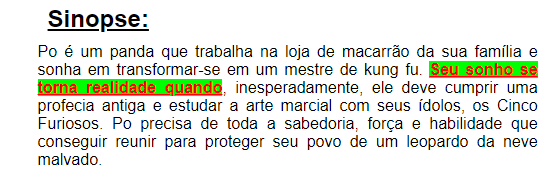
Este seletor básico irá escolher todos os nós. Ele também existe em um namespace único e em uma variante de todo o namespace também. Sintaxe: \* ns|\* \*|\* Exemplo: \* irá corresponder a todos os elementos do documento.

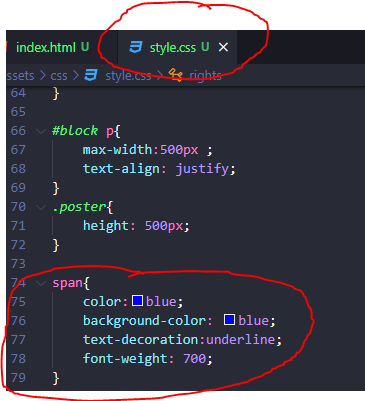
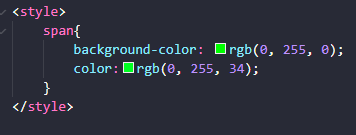
**Seletores por atributo**

Este seletor básico irá escolher nós baseados no valor de um de seus atributos, ou até mesmo pelo próprio atributo. Sintaxe: [atrib] [atrib=valor] [atrib~=valor] [atrib|=valor] [atrib^=valor] [atrib$=valor] [atrib\*=valor] Exemplo: [autoplay] irá corresponder a todos os elementos que possuirem o atributo autoplay (para qualquer valor).

Q5)Começando por ID, conforme foi ensinado nas aulas de HTML, o ID é um atributo que deve ser único no documento HTML, portanto, quando a estilização deve ser aplicada a um elemento apenas, utiliza-se seletor por ID (#valor-ID{}), isso garante a unicidade. Caso o desenvolvedor deseja que diversos elementos que possuam a mesma tag sejam afetados, é preferível que se utilize seletor por tagName (tagName{}), isso garante que todos os elementos com aquela tag serão atingidos. É interessante utilizar o seletor por class (.class) quando deseja-e afetar um ou mais elementos, que podem possuir tags diferentes, mas possuem a mesma class. Caso deseja um elemento específico, é possível fazer uma combinação de seletores, por exemplo, div .nome{} implica que está querendo aplicar uma estilização a elementos que possuam classe “nome” e que estejam dentro de uma “div”.

Q6) Apliquei os três tipos de estilos css (inline, internal e external). Além disso, pude verificar situações de prioridade entre elementos pai e filho.





Q7) Considere estilizar um elemento usando inline, internal e external. O estilo que prevalecerá será o do inline. A ordem de precedência é inline, depois internal, por último vem o external. Como o CSS segue o padrão de cascata, quando dois estilos do mesmo tipo estão sendo aplicados ao mesmo elemento, prevalece o último. Por exemplo, se no início de um arquivo CSS adicionamos p{color:black; Font-size:20px} e no final adicionamos p{color:blue}. O parágrafo terá o texto na cor azul (última propriedade color que foi aplicada a ele) e tamanho de fonte 20px. Pesquisando encontrei o seguinte também: 1)Inline tem mais prioridade, 2)depois o seletor por ID tem mais prioridade, por fim 3) class, pseudo-class e atributos.

Q8) As propriedades que tenham disso definidas de forma “inline” prevalecerão. Caso não tenha sido definida nenhuma propriedade de forma “inline”, prevalecem as propriedades adicionadas por último de forma “internal”. Se somente propriedades que tenham sidas definidas de forma externa, prevalece o último valor de cada propriedade. Portanto, as propriedades que foram adicionadas por último pelo seletor de ID serão aplicadas, no entanto as propriedades que foram modificadas pelo seletor só vão ser aplicadas se não tiverem sido usadas nenhum “inline” ou “internal”.

Q9) O repositório foi atualizado.