

♡ (◡‿◡) ♡

info adicional

para fazeres a tua webpage/site

notas:

vamos falar de:

1. favicon
2. boxmodel
3. flexbox

os materiais destas apresentações foram feitos com a ajuda do aidan quinlan, cujo curso de handmade web inspirou esta oficina e os materiais de estudo ajudaram a construir os PDFs :)

# 1. favicon

o favicon é o pequeno ícon que aparece ao lado do título da página no separador do browser. o favicon tem 32x32px ou 16x16px de tamanho e está no formato .png, .gif ou .ico para funcionar correctamente.

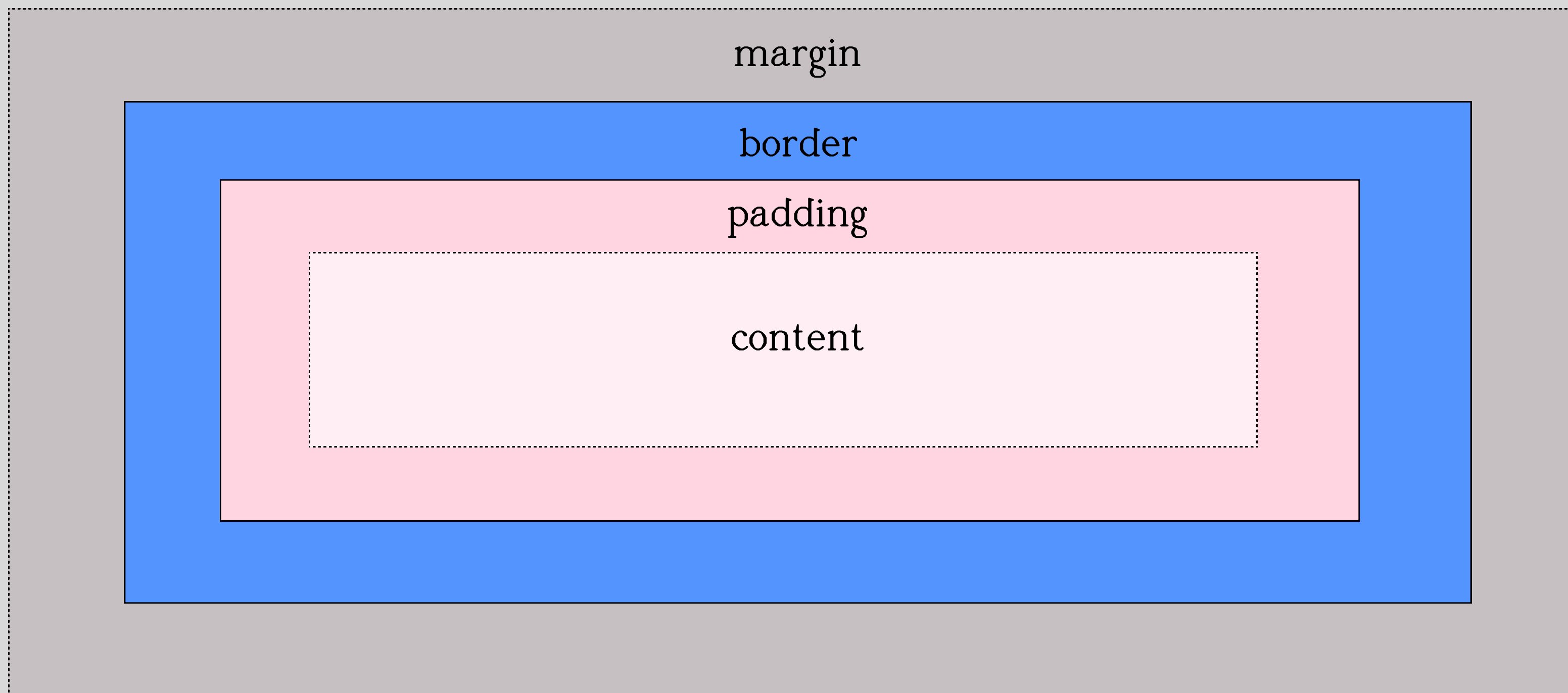
depois de teres uma imagem que cumpra estes requisitos, adiciona-a à pasta do teu website. de seguida, basta adicionar esta linha ao elemento <head> do teu documento HTML :) certifica-te de que ajustas o href e o tipo de ficheiro para corresponder ao teu.

```
<link rel=' icon' type=' image/png' href=' meuicone.png' >
```

## 2. box model

todos os elementos HTML numa página web são caixas retangulares, e compreender como são construídas é um passo importante para controlar o nosso website. isto é conhecido como o modelo de caixa (box model).

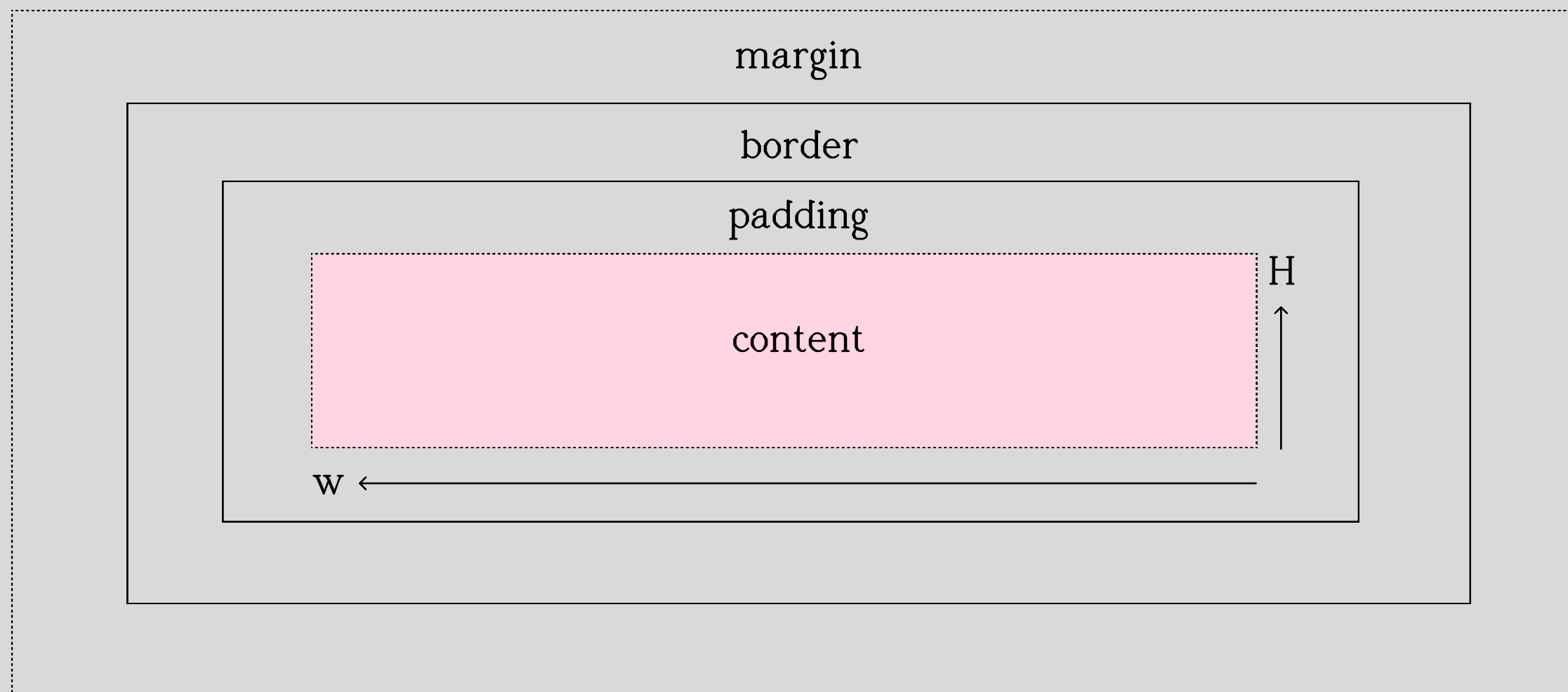
o modelo de caixa tem um aspeto semelhante a este:



## 2. box model - largura e altura

Podemos alterar o tamanho e o comportamento de cada caixa ajustando a largura (width), altura (height), preenchimento (padding), borda (border) e margem (margin) de um elemento.

ao definir um valor de largura (width) e altura (height) podemos controlar o tamanho da caixa de conteúdo. no entanto, isto não tem em conta os valores de padding, border e margin.



## 2. box model – medidas totais

então, para calcular a largura total (total width) de um elemento, precisaríamos de somar os valores de:

$$\text{margin-right} + \text{border-right} + \text{padding-right} + \text{width} + \text{padding-left} + \text{border-left} + \text{margin-left}$$

E para a altura total de um elemento:

$$\text{margin-top} + \text{border-top} + \text{padding-top} + \text{height} + \text{padding-bottom} + \text{border-bottom} + \text{margin-bottom}$$

Portanto, quando definimos os valores de largura e altura de um elemento, estamos apenas a definir os valores para a caixa de conteúdo e não para o tamanho total do elemento.

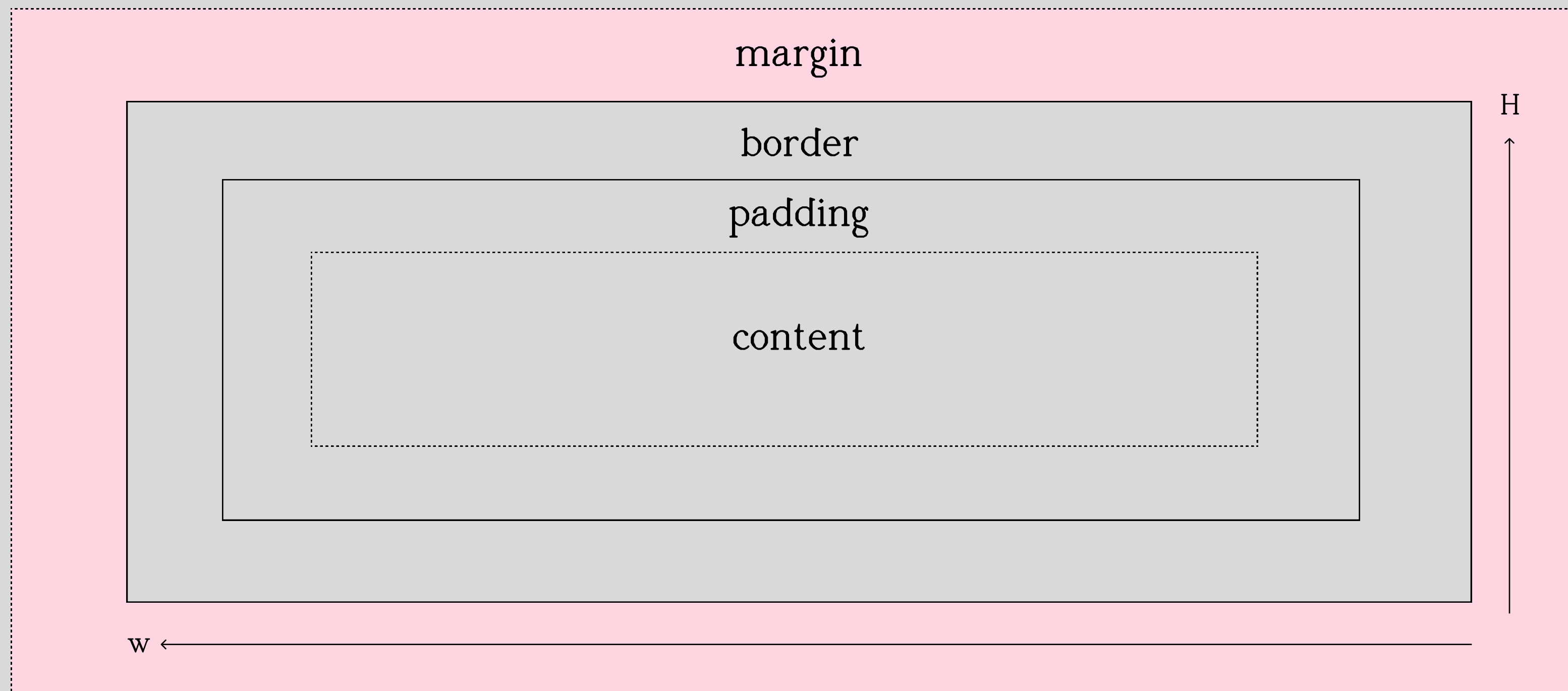
## 2. box sizing

podemos alterar a forma como a largura e altura total de um elemento são calculadas com a propriedade box-sizing. por defeito, a propriedade box-sizing está definida como content-box. Se alterarmos a propriedade box-sizing para border-box, podemos incluir os tamanhos do padding e da border na largura e altura total de um elemento.

```
box-sizing: border-box;
```

## 2. box sizing

Isto significa que quando definimos as propriedades de largura e altura de um elemento, este será sempre apresentado com essa largura e altura exatas, independentemente dos tamanhos do padding e da border.



## 2. box sizing

de um modo geral, este é o método de dimensionamento preferido pela maioria dos programadores porque simplifica o cálculo do tamanho real de um elemento. ao usar border-box, a única coisa que ainda precisa de ser considerada é a margem.

para facilitar as coisas, podemos adicionar esta regra de wildcard ao topo da nossa folha de estilos para que todos os elementos sejam apresentados usando border-box:

```
* { box-sizing: border-box; }
```

mais informação aqui:

- ~ Opening the Box Model
- ~ w3schools Box Model
- ~ w3schools box-sizing

## 3. flexbox

flexbox é um dos métodos que podemos usar para começar a criar layouts dinâmicos (flexíveis), grelhas e composições nas nossas páginas web. um flexbox tem duas partes básicas: o contentor flex (flex container) e os itens flex (flex items).

podemos transformar qualquer elemento num contentor flex atribuindo-lhe a propriedade display: flex;

todos os elementos contidos dentro deste elemento (os seus filhos) serão considerados itens flex. Usando o flexbox, podemos exercer um maior controlo sobre a posição, comportamento e alinhamento dos elementos contidos no interior. Os itens flex também podem receber display: flex; e tornar-se eles próprios contentores flex, permitindo-nos aninhar contentores dinâmicos dentro de contentores dinâmicos.

mais informação sobre flexbox neste [guia](#)

♡ (◡‿◡) ♡