

**CSS**



# CSS

## CASCADING STYLE SHEETS

DEFINES A WEB PAGE'S APPEARANCE

CSS SEPARATES STYLE AND CONTENT

CONSISTS OF A PLAIN TEXT FILE WITH RULES FOR THE DISPLAY OF HTML ELEMENTS

FORMATTING INCLUDES FONTS AND COLORS AS WELL AS LAYOUT AND POSITION

CAN BE CREATED OUTSIDE OF YOUR HTML AND APPLIED TO MULTIPLE WEB PAGES

WELL-FORMED HTML IS IMPORTANT FOR YOUR CSS TO WORK PROPERLY

# CSS

## HISTORY

**PRIOR TO CSS, WEB PAGES WERE COMMONLY STYLED WITH HTML TAGS AND STRUCTURED WITH TABLES**

**THIS WAS BOTH TEDIOUS AND INEFFICIENT**

**NINE DIFFERENT STYLE SHEET LANGUAGES WERE PROPOSED, TWO WERE CHOSEN AS THE FOUNDATION**

**CSS LEVEL 1 EMERGED AS A W3C RECOMMENDATION IN DECEMBER 1996**

**BROWSERS BEGAN TO SUPPORT CSS OVER THE NEXT FEW YEARS**

# CSS

## APPLICATION

**CSS CAN BE APPLIED IN THREE DIFFERENT WAYS TO A WEB PAGE:**

- **IN AN EXTERNAL .CSS FILE**
- **IN THE <HEAD> SECTION OF AN HTML DOCUMENT**
- **INLINE WITH HTML CODE**

# CSS

## STYLE CONSTRUCTION

### CSS RULE SET

**SELECTOR: INDICATES WHICH HTML  
ELEMENT WILL BE FORMATTED**

```
h1 {  
    color: green;  
    background: yellow;  
}
```

# CSS

## STYLE CONSTRUCTION

### CSS RULE SET

**SELECTOR:** INDICATES WHICH HTML  
ELEMENT WILL BE FORMATTED

**DECLARATION BLOCK:** DESCRIBES THE  
FORMATTING TO APPLY

```
h1 {  
    color: green;  
    background: yellow;  
}
```

# CSS

## STYLE CONSTRUCTION

### CSS RULE SET

**SELECTOR:** INDICATES WHICH HTML  
ELEMENT WILL BE FORMATTED

**DECLARATION BLOCK:** DESCRIBES THE  
FORMATTING TO APPLY

**PROPERTY/VALUE PAIR:** SPECIFIES  
FORMAT

```
h1 {  
    color: green;  
    background: yellow;  
}
```

# CSS

## STYLE CONSTRUCTION

### CSS RULE SET

**SELECTOR:** INDICATES WHICH HTML ELEMENT WILL BE FORMATTED

**DECLARATION BLOCK:** DESCRIBES THE FORMATTING TO APPLY

**PROPERTY/VALUE PAIR:** SPECIFIES FORMAT

**STYLE RULES ARE SEPARATED BY A SEMICOLON**

```
h1 {  
    color: green;  
    background: yellow;  
}
```



# CSS

## CASCADE

THE PRINCIPLE OF THE “CASCADE” IS APPLIED WHEN STYLE RULES ARE IN CONFLICT

THREE PRIMARY FACTORS DETERMINE WHICH STYLE RULE WINS OUT  
(IN ORDER OF WEIGHT):

**IMPORTANCE** - USE SPARINGLY IF AT ALL (EMERGENCIES)

```
color: green !important;
```

# CSS

## CASCADE

THE PRINCIPLE OF THE “CASCADE” IS APPLIED WHEN STYLE RULES ARE IN CONFLICT

THREE PRIMARY FACTORS DETERMINE WHICH STYLE RULE WINS OUT  
(IN ORDER OF WEIGHT):

**SPECIFICITY** - ELEMENT < CLASS < ID < STYLE ATTRIBUTE

```
h1{ color:green; }
```

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

```
#finished { color:blue; }
```

```
<h1 style="color:yellow;">TEXT</h1>
```

# CSS

## CASCADE

THE PRINCIPLE OF THE “CASCADE” IS APPLIED WHEN STYLE RULES ARE IN CONFLICT

THREE PRIMARY FACTORS DETERMINE WHICH STYLE RULE WINS OUT  
(IN ORDER OF WEIGHT):

**SOURCE ORDER** - IF ALL ARE EQUAL, LATER RULE IS USED

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

...

...

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

# CSS

## CASCADE

### OH YEAH...INHERITANCE

THE IDEA IS THAT SOME PROPERTY VALUES APPLIED TO AN ELEMENT WILL BE INHERITED BY THAT ELEMENT'S CHILDREN, AND SOME WON'T.

FONT-FAMILY AND FONT STYLINGS: INHERITED

MARGINS, PADDINGS, BORDERS AND MORE: NOT INHERITED

# CSS3

## CASCADE

**LATEST STANDARD FOR CSS**

**CSS2 IS BEST SUPPORTED**

**CSS3 IS STILL EVOLVING BUT OFFERS NEW FEATURES FOR DESIGNERS  
AND DEVELOPERS**

**MODERN BROWSERS SUPPORT MANY ASPECTS OF CSS3**

**BACKWARDS COMPATIBLE WITH CSS2**

# CSS3

## BEST PRACTICE

MOVE YOUR CSS STYLES INTO A COMPLETELY SEPARATE FILE, THEN LINK TO THE FILE IN THE <head> ELEMENT -

```
<head>
  <meta charset="utf-8">
  <title>Home Page</title>
  <link rel="stylesheet" type="text/css" href="styles.css">
  <link rel="icon" type="icon-x" href="favicon.ico">
</head>
```

### STYLES.CSS

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

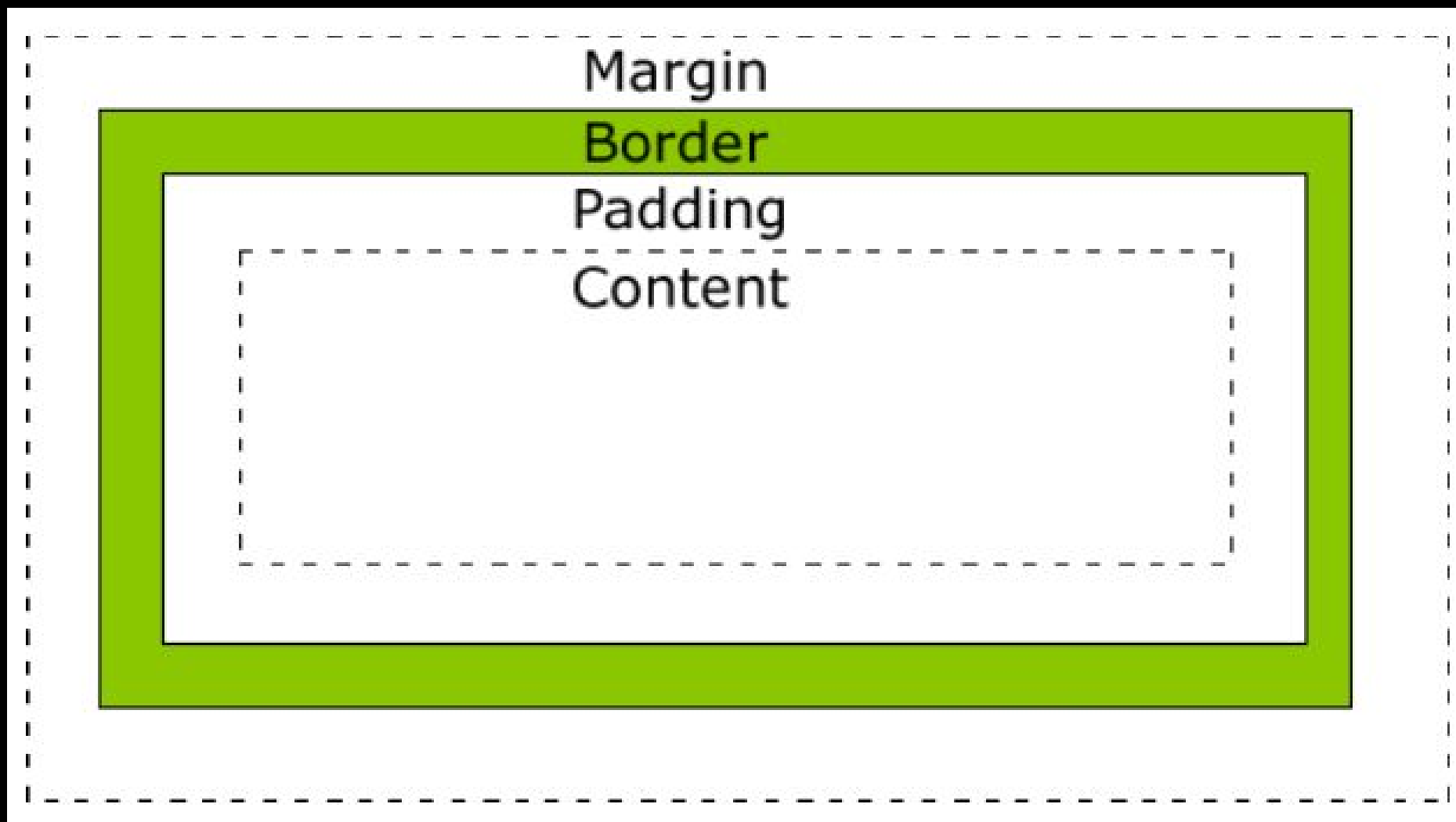
```
h1 {
```

# CSS

## CSS BOX MODEL

IN A WEB PAGE, EVERY ELEMENT IS RENDERED AS A RECTANGULAR BOX.

THIS BOX INCLUDES THE FOLLOWING, CHANGEABLE PROPERTIES.



# CSS

## CSS BOX MODEL

## SETTING AN ELEMENT'S WIDTH AND HEIGHT

- `auto` (default) Browser calculates the dimensions
- `Length` Define the dimensions in pixels or ems
- `%` Define the dimensions in percent of the container
- `inherit` inherited from the parent element



# CSS

## CSS UNITS OF MEASUREMENT

### ABSOLUTE MEASURE UNIT

**px**

```
body{  
  width: 1024px;  
  font-size: 16px;  
}
```

### RELATIVE MEASURE UNIT

**em** - relative to font-size of current element (normally 16px, so 1em = 16px;)

**%** - percentage of containing unit (area) or percentage of font-size of current

**vw** - viewport width (basically size of window or display, so 100vw = 100% width)

**vh** - viewport height

# **HOMEWORK**

**CSS ASSIGNMENT : DUE FEB 27**

**READ CHAPTERS 13 AND 14**