CSS

- Cascading Style Sheets
- encapsulate formatting / style
- style is defined through *properties*
- properties are name-value pairs

Applying Properties

- inline: `style` attribute of elements
    - *NOT RECOMMENDED*
- rules: selectors and properties
- `style` element in `head`
- external stylesheet

Advantages of CSS

- better separation of content and style
  - either as an attribute value
  - or as an element
  - or as a file

Degrees of Separation

- inline: as an attribute value
- valid only for the element
  - *NOT RECOMMENDED*

- internal: as an element
- valid for the document

- external: as external files
- valid for all importing documents

Inline Style

- single property:

```
<ELEMENT style="NAME: VALUE">
  ...
</ELEMENT>
```

- multiple properties

```
<ELEMENT style="NAME1: VALUE1;
                NAME2: VALUE2;">
  ...
</ELEMENT>
```

- *NOT RECOMMENDED*

Inline Style Problems

- limited separation of content and style
- clutters HTML code

Internal Element

- `style` element in `head`
- set style for any tag in the header:

```
<head>
  <style>
    TAG-NAME {
      NAME1: VALUE1;
      NAME2: VALUE2;
    }
  </style>
</head>
```

External Stylesheets

- separate file(s) for CSS
- `link` element in `head`

```
<head>
  <link rel="stylesheet" href="headings.css" />
  <link rel="stylesheet" href="tables.css" />
</head>
```

Serif vs Sans-Serif

- serif

  the quick brown fox jumps over the lazy dog

- sans serif

  the quick brown fox jumps over the lazy dog

Variable- vs Fixed-Width

- sans serif

  the quick brown fox jumps over the lazy dog

- monospace

  `the quick brown fox jumps over the lazy dog`

Font Usage

- serif
  - body text in print
- sans serif
  - most content of online documents
  - headings in print
- monospace
  - program listings

Font Family

- name: `font-family`
- value: list of fonts to try
- include a fallback

```html
<!DOCTYPE html>
<html>
  <head>
    <style>
      p { font-family: Cabin, Helvetica, sans-serif; }
    </style>
  </head>
  <body>
    <p>This font is sans-serif.</p>
  </body>
</html>
```

Font Size

- name: `font-size`
- value: absolute or relative size

Absolute Font Size

- units: `cm`, `mm`, `in`, `pt` (points), `pc` (picas), `px` (pixels)

```
<head>
  <style>
    p { font-size: 16pt; }
  </style>
</head>
<body>
  <p>This font has 16pt size.</p>
</body>
```

Relative Font Size

- units: `em`, `ex`, `%`, …

```
<style>
  body { font-size: 10pt; }
  p { font-size: 240%; }
</style>
...
<body>
  <p>This font has 24pt size.</p>
</body>
```

Font Description

- use many style rules at once:

```
<style>
  body { font-size: 6pt; }
  p {
    font-family: "Times New Roman", Times, serif;
    font-size: 3em;
  }
</style>
...
<body>
  <p>This font is 18pt serif and looks ugly on screen.</p>
</body>
```

Italics

- name: `font-style`
- values: `italic`, `oblique`, `normal`

```
<style>
  p { font-style: italic; }
</style>
...
<body>
  <p>This font is italic.</p>
</body>
```

Boldface

- name: `font-weight`
- values: `bold`, `regular`

```
<style>
  p { font-weight: bold; }
</style>
...
<body>
  <p>This font is bold.</p>
</body>
```

Font Property

- combine size, family and others
- name: `font`
- values: valid `font-` values in order

  style, variant, weight, size, family

```
<style>
  p { font: bold 16pt "Lucida Console", Courier, monospace; }
</style>
...
<body>
  <p>This font is bold 16pt monospace.</p>
</body>
```

Text Decoration

- name: `text-decoration`
- values: `underline`, `overline`, `line-through`, `none`

```
<style>
  p { text-decoration: underline; }
</style>
...
<body>
  <p>This text is underlined.</p>
</body>
```

Text Color

- name: `color`
- values: color names, hex values or rgb values

```
<style>
  p { color: pink; }
</style>
...
<body>
  <p>Floyd</p>
</body>
```

Background Color

- name: `background-color`

```
<style>
  p {
    background-color: #002855;
    color: white;
  }
</style>
...
<body>
  <p>Istanbul Technical University</p>
</body>
```

Text Alignment

- name: `text-align`
- values: `left`, `right`, `center`

```
<style>
  h1 {text-align: center;}
  p {text-align: right;}
</style>
...
<body>
  <h1>Istanbul Technical University</h1>
  <p>Pioneer through the ages</p>
</body>
```

List Bullets

- name: `list-style-type`
- values: `circle`, `square`, ...

```
<style>
  ul {list-style-type: circle;}
</style>
...
<ul>
  <li>Pink Floyd</li>
  <li>Deep Purple</li>
  <li>Black Sabbath</li>
  <li>White Lion</li>
</ul>
```

List Numbering

- name: `list-style-type`
- values: `upper-roman`, `lower-alpha`, ...

```
<style>
  ol {list-style-type: upper-roman;}
</style>
...
<ol>
  <li>Barış Manço ve Moğollar</li>
  <li>Cam Karaca ve Apaşlar</li>
  <li>Ersen ve Dadaşlar</li>
</ol>
```

Borders

- name: `border-style`
- values: `solid`, `dashed`, `dotted`, ...

```
<style>
  p {border-style: dashed;}
</style>
...
<p>
  In the beginning, the universe was created. This made a lot of
  people very angry, and has been widely regarded as a bad idea.
</p>
```

Border Width

- name: `border-width`
- values: `thin`, `thick`, size

```
<style>
  p {border-width: thin;}
</style>
...
<p>
  If it looks like a duck, and quacks like a duck, we have at
  least to consider the possibility that we have a small aquatic
  bird of the family anatidae on our hands.
</p>
```

Border Property

- combined width, style and color

```
<style>
  p {border: thin solid blue;}
</style>
...
<p>
  A computer terminal is not some clunky old television with a
  typewriter in front of it. It is an interface where the mind
  and body can connect with the universe and move bits of it
  about. (from Mostly Harmless)
</p>
```

Border Sides

- names: `border` `-top`, `-right`, `-bottom`, `-left`

```
<style>
  p {border-right: thin solid blue;}
</style>
...
<p>
  A common mistake that people make when trying to design
  something completely foolproof is to underestimate the
  ingenuity of complete fools.
</p>
```

Margins

- spacing outside the box
- name: `margin`
- value: size

```
<style>
  p {
    border-right: thin solid blue;
    margin: 2em;
  }
</style>
...
<p>
  You can't dodge your responsibilities by saying they don't
  exist.
</p>
```

Margin Sides

- names: `margin` `-top`, `-right`, `-bottom`, `-left`
- combined: top - right - bottom - left

```
<style>
  p {
    border-right: thin solid blue;
    margin-left: 2em;
  }
</style>
...
<p>
  When you blame others, you give up your power to change.
</p>
```

Padding

- spacing inside the box
- name: `padding`

```
<style>
  p {
    border-right: thin solid blue;
    padding: 1em 1.5em 0 1em;
  }
</style>
...
<p>
  Man has always assumed that he was more intelligent than
  dolphins because he had achieved so much —the wheel, New York,
  wars and so on-while all the dolphins had ever done was muck
  about in the water having a good time. But conversely, the
  dolphins had always believed that they were far more
  intelligent than man —for precisely the same reason.
</p>
```

Grouping Elements

- how to change the color of just one word / sentence?
- or any selection
- how to put a border around two paragraphs?

Group Level

- inline: `span`
- block: `div`

- no visible effect on their own

Inline Grouping

```
<p>
  The Beatles are regarded as
  <span>the most important and influential band</span>
  in the history of rock music.
</p>
```

- no visible effect, just a new inline element

Block Grouping

```html
<div>
  <p>Their famous lineup, called "The Fab Four",
    consisted of the following members:</p>
  <ul>
    <li>John Lennon (rhythm guitar)</li>
    <li>Paul McCartney (bass guitar)</li>
    <li>George Harrison (lead guitar)</li>
    <li>Ringo Starr (drums)</li>
  </ul>
</div>
```

- no visible effect, just a new block element

CSS Rules

- selectors for targeting elements:

```
SELECTOR {
   PROPERTY-NAME: PROPERTY-VALUE;
   PROPERTY-NAME: PROPERTY-VALUE;
   ...
}
```

CSS Selectors

- element name: all elements with a given name

```
em {
   font-style: regular;
   color: red;
}
```

Stylesheet Example

- `music.css`

```css
body {
  font: 16pt Roboto, Helvetica, sans-serif;
  background-color: bisque;
  color: rebeccapurple;
}

em {
  color: red;
  font-style: normal;
}
```

Selecting Classes

- how to apply same rule to multiple elements?
- *class* attribute
- multiple elements can have the same class

Class Example

- change colors of specific parts

```html
<p>
  The Beatles are regarded as
  <span class="info">
    the most important and influential band
  </span>
  in the history of rock music.
</p>
```

Class Styling

```
span.info {
  color: green;
}
```

- doesn’t have to be tied to an element:

```
.warning {
  color: red;
}
```

Selecting Specific Elements

- *id* attribute

```
<p>
  The Beatles are regarded as
  <span id="influence">
    the most important and influential band
  </span>
  in the history of rock music.
</p>
```

- multiple elements must NOT have the same id

Id Styling

```css
span#influence {
  text-transform: uppercase;
}
```

- or just:

```css
#influence {
  text-transform: uppercase;
}
```

Selectors

- selectors are used to select a subset of html elements
- expressions with simple pattern matching rules

Grouping selector

- separate each selector with a comma

```
/* This rule declares font-style as italic */
p, a {
  font-style: italic;
}
```

Grouping selector

```html
<!-- Which parts will be shown in italics? -->
<div>
  <p>Their famous lineup, called "The Fab Four",
    consisted of the following members:</p>
  <ul>
    <li>John Lennon (rhythm guitar)</li>
    <li>Paul McCartney (bass guitar)</li>
    <li>George Harrison (lead guitar)</li>
    <li>Ringo Starr (drums)</li>
  </ul>
  <a href="photo.html">See a group photo</a>
</div>
```

Descendant selector

- place the elements in hierachial order with a space

```
div p {
  font-style: italic;
}

section.note{
  color: red;
}
```

Descendant selector

- Which line(s) will be italic?

```html
<p>There are three formal models of complete computation</p>
<div>
  <ul>
    <li>&lambda;-calculus</li>
    <li>&mu;-recursive functions</li>
    <li>Turing machine</li>
  </ul>
  <p>Each one can realize any machine-computable task</p>
  <section class="note">
    <p>Each of the three models can be substituted
    with each other</p>
  </section>
</div>
```

Child selector

- place the related elements in hierachial order with `>`

```
div > p {
  font-style: italic;
}

section.note{
  color: red;
}
```

Child selector

- Which line(s) will be italic?

```html
<p>There are three formal models of complete computation</p>
<div>
  <ul>
    <li>&lambda;-calculus</li>
    <li>&mu;-recursive functions</li>
    <li>Turing machine</li>
  </ul>
  <p>Each one can realize any machine-computable task</p>
  <section class="note">
    <p>Each of the three models can be substituted
    with each other</p>
  </section>
</div>
```

Sibling selector

- place the immediately following sibling element with `+`

```
ul + p {
  font-style: italic;
}

#note{
  color: red;
}
```

Sibling selector

- Which line(s) will be italic?

```
<p>There are three formal models of complete computation</p>
<ul>
  <li>&lambda;-calculus</li>
  <li>&mu;-recursive functions</li>
  <li>Turing machine</li>
</ul>
<p>Each one can realize any machine-computable task</p>
<p id="note">
  Each of the three models can be substituted with each other
</p>
```

General sibling selector

- place the following sibling elements with `~`

```
ul ~ p {
  font-style: italic;
}

#note{
  color: red;
}
```

General sibling selector

- Which line(s) will be italic?

```
<p>There are three formal models of complete computation</p>
<ul>
  <li>&lambda;-calculus</li>
  <li>&mu;-recursive functions</li>
  <li>Turing machine</li>
</ul>
<p>Each one can realize any machine-computable task</p>
<p id="note">
  Each of the three models can be substituted with each other
</p>
```

Pseudo-classes

- A set of pseudo-classes are defined for element states
- E.g. `a:hover`, `div:first-child`

```css
p.note, li:first-child {
  font-style: italic;
}

p.note:hover {
  background-color: yellow;
}
```

Pseudo-classes

- Which line(s) will be italic?
- Which line(s) change color on mouse hover?

```
<p>There are three formal models of complete computation</p>
<ul>
  <li>&lambda;-calculus</li>
  <li>&mu;-recursive functions</li>
  <li>Turing machine</li>
</ul>
<p>Each one can realize any machine-computable task</p>
<p class="note">
  Each of the three models can be substituted with each other
</p>
```

Pseudo-elements

- Pseudo-elements are used to stylise element parts
- E.g. `::first-line`, `::before`, `::after`

```css
/* fix ek$i */
p::first-letter {
  text-transform: uppercase;
}

p.note::after {
  content: "!!!";
}
```

Pseudo-classes

- How does the following render?

```html
<p>there are three formal models of complete computation</p>
<ul>
  <li>&lambda;-calculus</li>
  <li>&mu;-recursive functions</li>
  <li>turing machine</li>
</ul>
<p>each one can realize any machine-computable task</p>
<p class="note">
  each of the three models can be substituted with each other
</p>
```

Attribute selectors

- Selects elements based on attributes

```
p::first-letter, li[id="upper"]::first-letter{
  text-transform: uppercase;
}

p[class]::after {
  content: "!!!";
}
```

- Also check `~=`, `|=`, `^=`, `$=`, `*=`

Attribute selectors

- How does the following render?

```html
<p>there are three formal models of complete computation</p>
<ul>
  <li>&lambda;-calculus</li>
  <li>&mu;-recursive functions</li>
  <li id="upper">turing machine</li>
</ul>
<p>each one can realize any machine-computable task</p>
<p class="note">
  each of the three models can be substituted with each other
</p>
```

CSS variables

- CSS variables starts with `--`
- Use `:root` pseudo-class to define global variables

```css
/* Define colors with CSS functions */
/* rgb, rgba, hsl, hsla */
:root {
  --itublue: #002855;
  --itugold: rgb(151, 128, 79); /* #97804f */
  --ituskyb: hsla(193, 48%, 62%, 0.5); /* #7bafd4 w/ 0.5 opacity */
}
```

CSS variables

- Use the variable values with `var()` function

```css
p {
  background-color: var(--itublue);
  color: var(--itugold);
}
p:hover {
  opacity: 0.5;
}
:not(p){
  background-color: var(--ituskyb);
}
```

CSS variables

- How does the following page render?

```html
<body>
  <p>Istanbul Technical University</p>
</body>
```