

# HTML & CSS: Text documents, forms, formatting and layout

Michał Okulewicz

Wydział Matematyki i Nauk Informatycznych  
Politechnika Warszawska

# Lecture plan

- 1 Motivation for today's lecture
- 2 HTML
  - Introduction
  - Standard document elements
  - Forms and controls
  - HTML5 interactive elements
- 3 CSS
  - Introduction
  - CSS formatting
  - CSS layout
  - Clean CSS code
- 4 More details

# Web applications vs. desktop applications

TECH / 2012 TECH

## 2012: The Year The Desktop App Died

**We've seen the last of the great desktop apps.** The end of one of computing's longest eras.



**John Herrman**  
BuzzFeed Staff

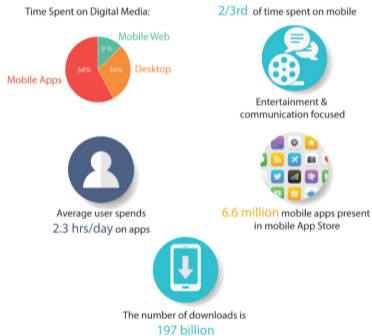
Posted on December 12, 2012, at 10:47 a.m. ET

Source: <https://www.buzzfeednews.com/article/jwherrman/2012-the-year-the-desktop-app-died>

Discussion: <https://www.sitepoint.com/web-desktop-apps/>

# 2017 status of mobile web applications and their future

## MOBILE APP STATS 2017



Source: [Future of Mobile App Development 2018](#)  
**Progressive Web Apps** as possible future trend

# HTML

- Purpose and structure of HTML
- HTML syntax and its connection to XML
- Block and inline elements
- Functional and visual elements
- Specification and tutorial examples

# Empty HTML document

```
<!DOCTYPE html>
<!-- DOCTYPE html declares document to be HTML5 -->
<html>
<head>
  <meta http-equiv="content-type"
        content="text/html; charset=utf-8" />

  <!--
  New syntax for the above declaration:
  <meta charset="utf-8" />
  -->
  <title>Page title</title>
</head>
<body>
  <!-- The contents of the page goes here -->
</body>
</html>
```

Checkout how to  
declare older formats

As usual: BEWARE  
of older browsers!

## HTML vs. XHTML

No attribute values

No closing "/>" tag

`<input type="checkbox" checked>` - an HTML style checkbox

`<input type="checkbox" checked="checked" />` - an XHTML style checkbox

### Comment

HTML 5 is descendant of both HTML 4.0.1 and XHTML 1.1

# Basic text document

## Some quotes from “The Lord of the Rings”

### Life

It's a dangerous business, Frodo, going out your door. You step onto the road, and if you don't keep your feet, there's no knowing where you might be swept off to.

### Afterlife

PIPPIN: I didn't think it would end this way.

GANDALF: End? No, the journey doesn't end here. Death is just another path, one that we all must take. The grey rain-curtain of this world rolls back, and all turns to silver glass, and then you see it.

PIPPIN: What? Gandalf? See what?

GANDALF: White shores, and beyond, a far green country under a swift sunrise.

PIPPIN: Well, that isn't so bad.

GANDALF: No. No, it isn't.

Headers

Paragraphs

(Note to self: `html_examples/headers_paragraphs.html`)



## Some content element examples

Element	Function
html	Root of the document structure
head	Metadata of the document and HTTP headers
body	Visible contents of the document
p	Text paragraphs
h1-h6	Various headers
br	New line (break line)
a	Hyperlinks
img	Images

## Links and images

a

- relative links:

```
href="../downloads/mlog/01_fake_algorithm_run.zip"
```

```
href="/~okulewicz/downloads/mlog/01_fake_algorithm_run.zip"
```

- absolute links:

```
href="http://www.mini.pw.edu.pl/~okulewicz/downloads/mlog/01_fake_algorithm_run.zip"
```

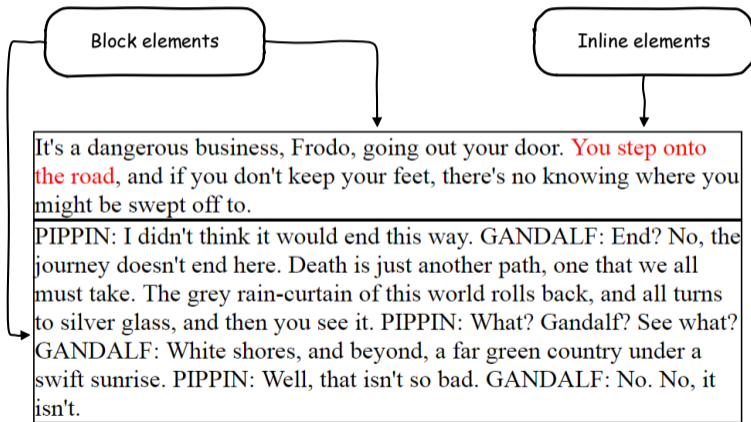
img

- src - image source
- alt - alternative text (useful for visually impaired, text browsers and search engines) - [further reading](#)

# Table structure

Element	Function
<code>table</code>	Tables
<code>tbody</code>	Main content of the table
<code>thead</code>	Headers of the table
<code>tfoot</code>	Footer of the table
<code>tr</code>	Row of the table
<code>td</code>	Single data cell of the table
<code>th</code>	Single header cell of the table

## Two groups of elements



## Contextual elements examples

Element	Function
<code>div</code>	Generic block element
<code>section</code>	Document section
<code>nav</code>	Navigational bar
<code>p</code>	Text paragraph
<code>span</code>	Generic inline element
<code>strong</code>	A marked part of the text
<code>em</code>	An emphasized part of a text
<code>del</code>	A part of the text that has been removed (or obsolete)
<code>small</code>	Side comments

Note (from HTML 5.2 W3C Recommendation, 14 December 2017)

"Authors are strongly encouraged to view the `div` element as an element of last resort, for when no other element is suitable."

# Form and controls

- In order to develop web applications (either client-based or full-stack), we need a set of controls to get an input from the user.
- Before HTML5 only the basic ones (text, password, checkbox, lists etc.) has been natively supported by browsers
- HTML5 gave additional support for autocompletion, validation, and more control types

# Functional elements

Element	Function
form	Container marking a context for a single data submit operation
input	Text, password, checkbox controls
select	ComboBox and list controls
textarea	Large blocks of texts control
button	Buttons :)

## Visual and user experience elements

Element	Function
<code>fieldset</code>	Visually groups controls
<code>legend</code>	Label for a fieldset
<code>label</code>	Label for a single control (enlarges active area for clicking and tapping)
<code>data</code>	A data list for initial autocomplete



# Form example

Bank transfer details

Account number

Recipient

Transfer note

Amount

Transfer date

Transfer type  ELIXIR  SORBNET

Additional details

Buy insurance for the transfer

Confirmation e-mail

(Note to self: [html\\_examples/forms.html](http://html_examples/forms.html))

# HTML5 – more than just new semantic elements

- Audio & video
- Canvas
- Drag & drop operations

Please checkout usage of canvas and audio elements

## Final message

Don't be afraid of **HTML specs**  
Checkout **list of tags on w3schools**

# CSS

- Purpose of CSS
- CSS basic syntax
- CSS formatting
- CSS layout
- Media queries

# Why CSS?

- You can do quite a lot basing on default formatting of various HTML elements
- Lets inspect some of them...
- (Note to self: [html\\_examples/formatting-no-css.html](http://html_examples/formatting-no-css.html))
- But not with interface design and its layout
- (Note to self: [html\\_examples/formatting.html](http://html_examples/formatting.html))
- Moreover, one should not rely on such a default formatting

# Why CSS?

- You can do quite a lot basing on default formatting of various HTML elements
- Lets inspect some of them...
- (Note to self: `html_examples/formatting-no-css.html`)
- But not with interface design and its layout
- (Note to self: `html_examples/formatting.html`)
- Moreover, one should not rely on such a default formatting

# Why CSS?

- You can do quite a lot basing on default formatting of various HTML elements
- Lets inspect some of them...
- (Note to self: `html_examples/formatting-no-css.html`)
- But not with interface design and its layout
- (Note to self: `html_examples/formatting.html`)
- Moreover, one should not rely on such a default formatting

# Why CSS?

- You can do quite a lot basing on default formatting of various HTML elements
- Lets inspect some of them...
- (Note to self: `html_examples/formatting-no-css.html`)
- But not with interface design and its layout
- (Note to self: `html_examples/formatting.html`)
- Moreover, one should not rely on such a default formatting



## CSS syntax

```
selector {  
  style: value;  
}
```

Selector	Function
element	Selects all: <element>
.className	Selects all: <element class="className">
#identifier	Selects all: <element id="identifier">
:pseudo-class	Selects all with context (e.g. :first-of-type)
[filter=value]	Selects all: <element filter="value">

## CSS in text formatting

**color** foreground content of the selected elements (usually: text, possibly: underline etc.) - named `color`, `#34DE12` or `rgb(100,200,123)` syntax possible

**background** complex style including background color and image and its repetitiveness

**border** complex style including color, width and type of border (e.g. `solid`, `dashed`)

**font** font family (including named and generic: `serif`, `sans-serif`, `monospace`, `cursive` and `fantasy`), size, weight and style

**text** alignment, decorations and transformations

## CSS in layout design

**position** way of setting elements positions (static, relative, absolute, fixed)

**display** way of rendering elements

**float** makes an element a floating element (it will be surrounded by other content)

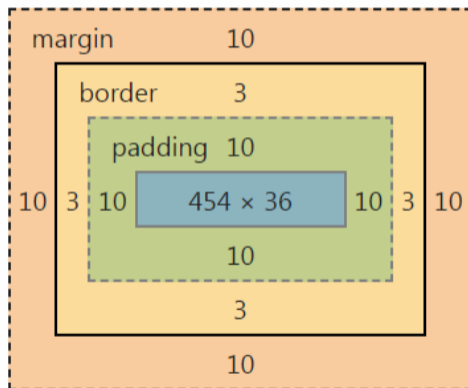
**left—top—right—bottom** position properties

**margin** element's margins (starting from top and going clockwise)

**padding** element's inner margins (starting from top and going clockwise)

(Note to self: larger layout example [html\\_examples/ip\\_3\\_css.html](http://html_examples/ip_3_css.html))

# Box model



# Grid model

Bank transfer details

Account number	<input type="text"/>
Recipient	<input type="text"/>
Transfer note	<input type="text"/>
Amount	100
Transfer date	rrrr-mm-dd

Send money nowhere

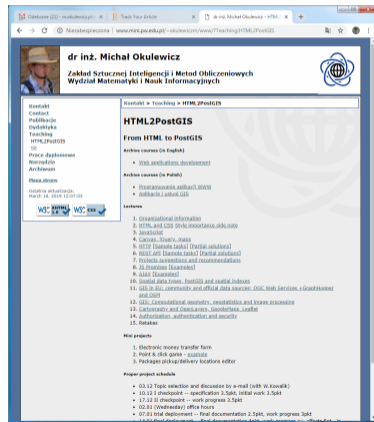
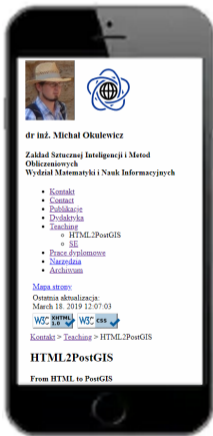
```
.grid-container {  
  display: grid;  
  grid-template-columns: 200px auto;  
}
```

```
<div class="grid-container">  
  <div><label for="number" class="label-text">Account number</  
  <div><input id="number" class="input-transfer-data" name="nu  
  <div><label for="recipient" class="label-text">Recipient</la  
  <div><input id="recipient" class="input-transfer-data" name=  
  <div><label for="note" class="label-text">Transfer note</lab
```

# Media queries and Responsive Web Design

- How to create web interfaces simultaneously supporting:
  - standard browsers (on +15" screens)?
  - mobile devices (with 7" – 11" screens)?
  - smartphones (with 4" – 6" touchscreens)?

# Media queries and Responsive Web Design



## Small device query example

```
@media (max-device-width: 640px) {  
  body {  
    background: white;  
  }  
  div {  
    border: 0px solid #003399;  
    position: static;  
    background: #ffffff;  
    margin: 5px;  
    color: #000000;  
    width: 90%;  
  }  
}
```

Beware: A pixel is not a pixel



# Clean code

- CSS preprocessors: LESS, SASS
- Flat structure vs deep structure
  - [Block Element Modifier](#)
  - Utilities (.u- prefix)
  - JS Hooks (.js- prefix)

## Example of a pointed HTML & CSS task (1)

- General:
  - 3p. For an (almost) complete solution valid with HTML5 and CSS 3 standards
- General layout:
  - 1p. Create a three column display with each of the columns taking 30
  - 1p. Paragraphs start at the same level as the caption of the form
  - 1p. Paragraphs have a 1em sized side margins
  - 1p. Title bar displays a provided icon and "Thursday 10:15 a.m." text
- General layout for browser width less than or equal 1080px:
  - 2p. The display switches to single column, with the form coming first
  - 1p. The form is centered and has a 20em width

## Example of a pointed HTML & CSS task (2)

- Text content:
  - 1p. Justify the text in all the columns.
  - 1p. The right and left columns present 5 paragraphs of text (e.g. from lipsum.com).
- Registration form features:
  - 1p. Only alphanumeric characters allowed in login
  - 1p. Passwords at least 8 signs long
  - 1p. All fields are set with an initial text: "Only alphanumeric characters", "Please enter e-mail", "At least 8 characters"
  - 1p. Login and passwords are required fields

## Example of a pointed HTML & CSS task (3)

- Registration form display:
  - 1p. Registration form with proper types of input fields and usage of labels
  - 1p. Form has a rounded border
  - 1p. Fields are separated by a 0.5em distance
  - 1p. Pink background of an invalid input and green background of a valid input
  - 1p. Required field has a red border
  - 1p. Input takes 50
  - 1p. Labels are aligned to the right with 1em distance between them and the inputs
  - 1p. Buttons are centered
  - 1p. Every second input entry has a blue background

## Example of a pointed HTML & CSS task (large viewscreen)

The screenshot shows a web browser window with the address bar displaying the file path: file:///C:/Users/okulewicz/Documents/Dydaktyka/WWW/html\_css/zaliczenie/2016\_17/index.html. The page content includes a registration form and several paragraphs of Lorem Ipsum text.

**Registration form**

login	<input type="text" value="Only alphanumeric characters"/>
e-mail	<input type="text" value=""/>
password	<input type="password" value="At least 8 characters"/>
confirm password	<input type="password" value="At least 8 characters"/>

The form is styled with a rounded border and a light blue background. The input fields have a light green background. The validation messages are displayed in red text within the input fields. The text on the page is styled with a serif font and justified alignment.

# Example of a pointed HTML & CSS task (small viewscreen)

