# HTML/CSS Udemy Course

## Section 2

HTML: Language that web pages are written in.
HTML = Hyper Text Mark-up Language
<p>content<\p>, <p> = element tag
Language used to describe content
CSS NOT for style/design
HTML ONLY for content

Nested elements, bulleted list
<ul>
    <li>milk</li>
    <li>milk</li>
</ul>

Content goes inside body, always start with <html> element

Head element, information about the html page, for example title, the title will appear as the name of the tab but won't appear in the content of the web page
Also include character encoding information — what set of characters do I want to have access to? For English speaking audiences, utf-8 makes sense

<!DOCTYPE html> = most recent version of the html language

Link: <a href="link"></a>
href = hypertext reference

Summary:
- HTML is everywhere!
- Pages contain elements
- Russian stacking dolls
- Document structure
- Attributes (href)

## Section 3

Include an image: <img src="address of your image file" alt="description of the image in case it doesn't open">
No need to close with </img>

Include sound:
<audio controls>
<source src="audio/test.mp3" type="audio/mpeg">
<source src="audio/test.ogg" type="audio/ogg">
</audio>
When you include different types of files, it will choose the one that is supported. You can add fallback messages.

**Think about the browsers that your users will use!!!**

When it comes to video, the real world is a little more complicated because there are more factors to consider: browser versions, video resolution, custom ui/ branding concerns (must use javascript)
All of these considerations fall outside the scope of html
Solution — Video hosting services

## Section 4

Use headings to create hierarchy and keep text organized.

h1, h2, h3, ..., h6

Only use as many headings as you need.

There are many list types:
1. Unordered list <ul> (bulleted list)
2. Orderest list <ol> (numbered list)
3. Description <dl>, <dt> and then <dd>

Bold =  (as opposed to <b>)
Italic =  (as opposed to <i>)

And then you can personalize strong/em using CSS.

For definitions you could also use <dfn>.

There is an attribute called class where you can create your own element and then use CSS to stylicize that class.

There are special characteristics that have codes.

## Section 5

Structural elements house other elements.

<header></header>
<footer></footer>
<article></article>
<aside></aside> related to the content surrounding it but should not be IN it, for example a glossary

You can use these in the page overall but also within sections.

<nav></nav> = navigation, house navigation links, user hops around from page to page
Might have something like:
<nav>
    <li><a href=“”></a>
    <li>…
<\nav>

divide page into chunks
Used if no semantic element works of your purpose

When you break up your HTML, doing CSS becomes a lot easier.

 non-semantic version of  or 
Used if you want to style something for marketing reasons, maybe it should be italic but in a different way than a definition word. Maybe it’s a brand name.

HTML comment = <!- - your comment goes here - -> (there is no space between the dashes)

Include comments when the div **ends** <! - - /article-body - ->

<section>think of this as chapters</section>
You can nest one section into another.

You can use h1 everywhere and then create sub-sections instead of h1, h2, h3…
Sectioning is another way to create a hierarchy.

## Section 6

Forms: First name, Last name, Password for example
<form>

```
    <label for="firstName">First Name:</label>
    <input id="firstName" type="text">
</form>
```

name="firstName", see form
Name attribute helps for programming (with Python for example

There are many types for forms, it's good to use the right one because it helps the user to know they have to enter an e-mail and because the form won't submit if the input is not of the right type.

There's "email", "tel",

You can allow the user to type a whole paragraph.

Form fields:
1. Select (dropdown)
2. Radio (choose one)
3. Checkbox (check all that apply)

## Section 7

Table of data: tr=table row, td=table data/column, th=table header

Do not use tables for layout!!!!!!!!! Use CSS instead!!!!!!!!

## Section 8

CSS = Cascading Style Sheets

HTML: describe content
CSS: describe presentation

Rel = relationship, stylesheet

```
<link rel="stylesheet" href="css/screen.css">
```

This is a CSS rule:
```
h1 {
    color: orange; (this is a declaration)
}
```

CSS Selectors:

1. Type selectors
2. Descendant selectors
3. Class selectors

Type selectors: targets a type, for example h1

Descendant selectors: for example paragraphs of headers, header p or header p span

Class selectors: give a class to a paragraph for example, <p class=”highlight”>, then

.highlight {
Background-color: yellow
}

You can even mix in descendent and class for example header .highlight {}

Cascade
- Water == stylistic instructions

Waterfall from top to bottom
- Browser default
- Our CSS
- Inheritance
- Specificity: which instructions to listen to if some properties are competing? Choose most specific one. Class is also more specific than type.

The box model: Every element a page is a collection of boxes.

Every element has properties such as padding, border

<div class

A parent div is not aware of its floated children.

"content-area group" multiple classes

“Content fix” code to paste:

```
.group:before,
.group.after {
      content: "";
```

```
    display: table;
}

.group:after {
    clear: both;
}

.group {
    zoom: 1;
}
```

The float at width properties can be used to create columns but make sure you CLEAR your floats!

Use % and or px to manage sizes.

Use HTML Color Picker to find colors.

```
    padding-left: 40px;
    padding-right: 40px;
    padding-top: 20px;
    padding-bottom: 20px;
```

OR

```
    padding: 20px 40px 20px 40px;
```

OR

```
    padding: 20px 40px;
```

## Section 9

css comments: /* */

Position: relative;
Position: absolute; aka. “Overlapping effect”
Position allows to create x and y coordinates.

Reba = red green blue alpha-transparency, a=0 completely transparent, a=1 completely opaque

If two things have position: relative; the one that comes latest in the html code will

be on top.

Z-index: whichever element has the highest z-value will be placed on top of the other elements. —> z-index == stacking order

## Section 10

Styling tables

Zebra striping - every other row

:nth-child() pseudo-selector

## Section 11

Font-family: Tahoma (1st choice of font), sans-serif (keyword);

Font-family: Helvetica, Arial, sans-serif;

“Times New Roman”

Font-size: px/%/small/medium/

Define a baseline size in pixels, and then use percentages to manage size of other elements in the page.

Font-weight: normal; (bold)
Font-style: italic;
Letter-spacing: 4px;
Word-spacing: 10px;
Text-shadow: (horizontal offset) (vertical offset) (blur radius) (color);
Text-transform: uppercase;
Text-decoration: none; (remove underlines from links)
Text-align: right/center/justify;
Text-indent: 25px;
Line-height: 1.5;

Custom Fonts: embedding font files: @font-face

1. Hosted font solutions
2. Self hosted fonts

(Self-hosted fonts)

Find on Font Squirrel
Webfont generator to convert a font into all formats
Create new folder, fonts and move over the fonts into your fonts folder

Into css file

*/ Custom fonts */

```
@font-face {
	font-family: 'PTSans';
	src: url('fonts/PT-Sans.woff') format('woff'),
		url('fonts/PT-Sans.eot') format('embedded-opentype'),
		insert url for all types
	font-weight: normal;
	font-style:normal;
}
```

Add font-face rules for italic, bold, etc.

## Section 12

Web browser's developer tool - inspect elements
Good for troubleshooting or testing things out
Great to track down where properties are coming from
Check out websites you like and see if there are things you want to emulate

## Section 13

**Background image:**

background: url(../images/color-tile.png) left bottom repeat-x, #FFF url(../images/color-tile.png) left top repeat-x;

By default images tile to fill up the whole element we're trying to color.

So you can add

Background-repeat: no repeat;

So we have

Background-image: url(../images/color-tile.png);

Background-repeat: repeat-x; (repeat horizontally)
Padding-top: 50px;

By default, background images align with the left side of the element they are aligned with. You can change that with:

Background-position: (horizontal) (vertical)
So for example: background-position: right center;

And then background-repeat: no-repeat

You can resize images: background-size: cover;

Shortcut:

**Background: #c2bbb1 url(../images/bird.jpg) center center no-repeat;**

You can create multiple backgrounds by using the comma, they will come in order. Always provide a backup color.

**Background gradient:**

Background-image: linear-gradient(to bottom, #e5e9dc, #FFF)
Background-image: linear-gradient(to right/to bottom right, #e5e9dc, #FFF)

Use background-color: #FFF; so that in case the web browser can't load the gradient background, there's a fallback color

Sprite is a file with a few images in it. It's faster than having different images. You can use it to create your own icons for example.

"Icon art", check out folder

Use gimp.org to create your icons, online photoshop tool

Make sure to space your elements equally, for example each of them 50px*50px

**Full width background:**

The background area should have unlimited maximum width but you have to make sure the header elements, among other elements, need to have a fixed max-width value.

## Section 14

Responsive web design
Very important for user experience
Needs to work flawlessly for all kinds of design

Recipe:

1. Meta tag in <head> - the device should stay true to its self
2. Media queries (**conditional CSS**) - create conditions that only work for certain dimensions

For example,
@media screen and (max-width: 700px) {

.main-area, .sidebar {
Width: auto;
Float: none;
}

}

class="hide-small" to hid elements that should disappear once small

Responsive Grids:

A grid is a structure that helps web designers. 12-column grid is the most popular.

For mobile, usually one element per row.

Xs = extra-small (smart phone)
Sm = small (tablet)
Md = medium (laptop)
Lg = large (desktop computer)

For example, col-lg-3 (12/4=3 so 3 is how many sections there are)

## Section 15

CSS Special Effects:

There is a box-shadow property. There is also the inset, which is an inside shadow.

Rounded corners: border-radius (takes in 4 values, clock-wise from top etc.)

There are transformations.

There are also animations.