

# Colors/Tables

CSCI 1210

Essentials of Web Design

# CSS Review

# Review

HTML == Structure

CSS == Style

We use CSS to modify the presentation of our web pages

The preferred method of including CSS with our HTML is through use of External stylesheets

For special situations, however, we can use Embedded (whole page) or Inline (specific element) CSS

# Review

PROPERTY

VALUE

SELECTOR

The diagram illustrates the components of a CSS rule. A red box highlights the selector 'p', with a red arrow pointing to it from the label 'SELECTOR'. Another red box highlights the property names 'width:', 'height:', 'margin:', 'padding:', and 'font-weight:', with a red arrow pointing to it from the label 'PROPERTY'. A third red box highlights the values '100px;', '75px;', '50px auto;', '25px;', and '200;', with a red arrow pointing to it from the label 'VALUE'. The entire CSS rule is enclosed in a red box.

```
p {  
  width: 100px;  
  height: 75px;  
  margin: 50px auto;  
  padding: 25px;  
  font-weight: 200;  
}
```

# Review

Three parts:

Selector -> Identifies (or “selects”) the element you want to style

Property -> What you want to change (e.g., “margin”)

Value -> How you want to change it (e.g., 25px) [px == pixels; it is important that you don't put a space between 25 and px]

# Review (OK, this is actually new)

So “px” is pixels

There are other units of measure we can use

- % - percentage
- em - literally, the width of the letter ‘m’
- rem - font size of the root element
- vw - 1% of the viewport width
- vh - 1% of the viewport height
- cm - centimeters
- in - inches
- pc - picas
- pt - points

In this class, we’ll mostly use px

It is an absolute length unit that displays the same size regardless of the display size

Other units (e.g., em and rem) display differently on different sized displays (more on that in CSCI 1720)

# More on CSS & Color

Choosing Colors

# Choosing Colors

As we learned from our first homework assignment, poor choice of display colors can ruin an otherwise good web site, and potentially a visitor's eyesight

All joking aside, a badly executed color scheme will encourage users to find other sites

Several online tools are available that can help a developer select an effective set of colors for a web site (this, by the way, may actually be decided by the client)

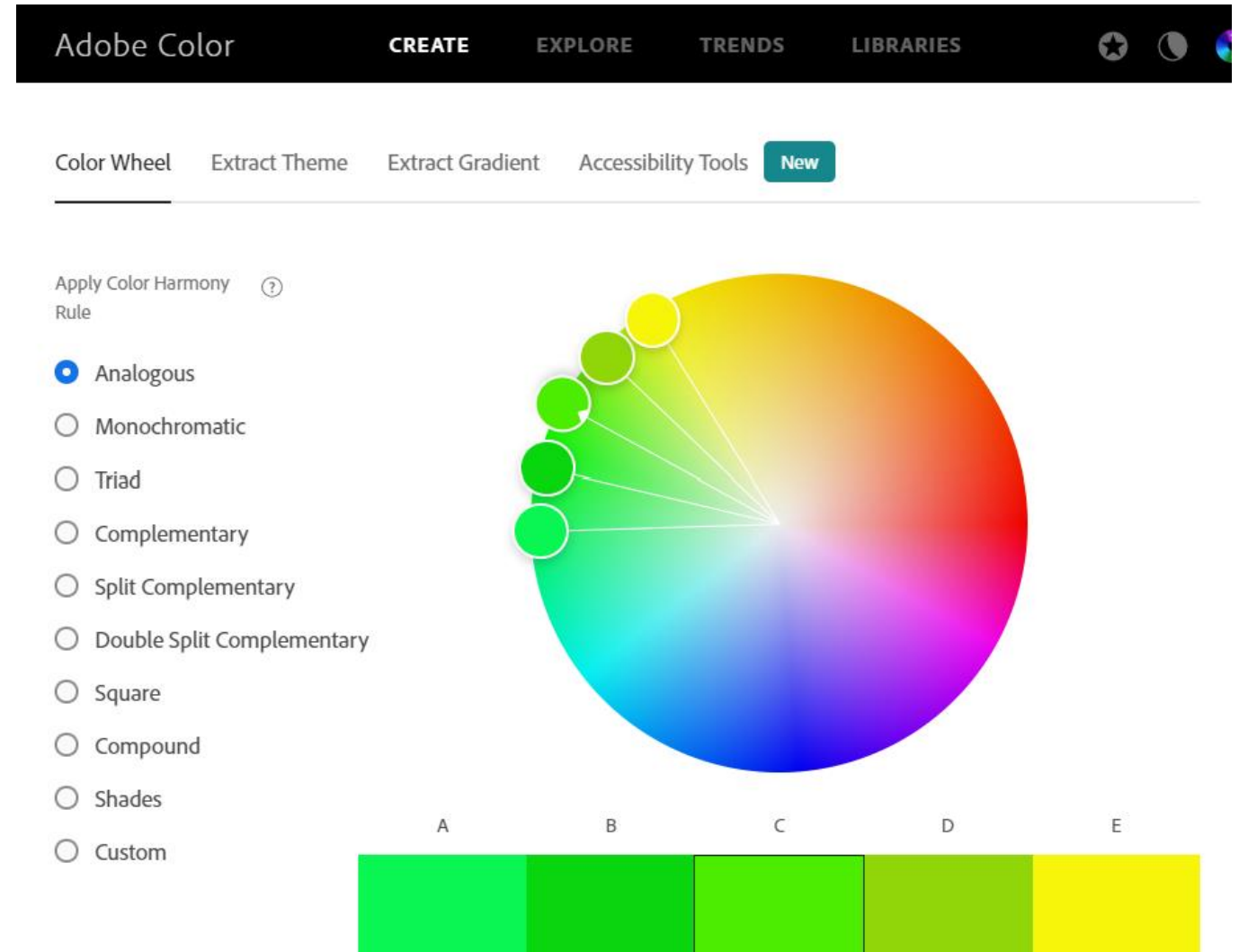
(If you were worried, you won't have to do hexadecimal math)



# Choosing Colors

[Adobe Color Wheel CC](https://color.adobe.com/create/color-wheel/)

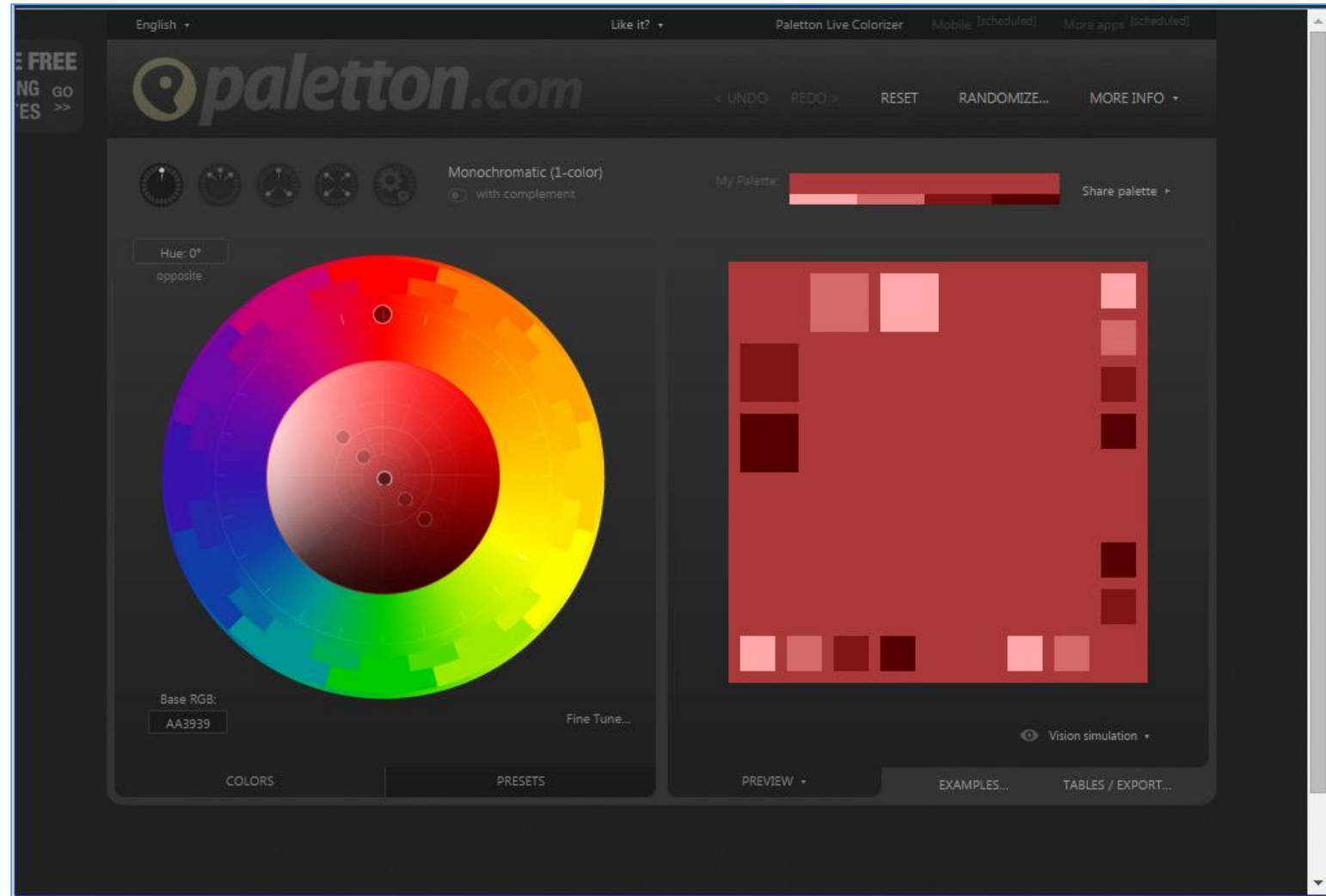
<https://color.adobe.com/create/color-wheel/>



# Choosing Colors

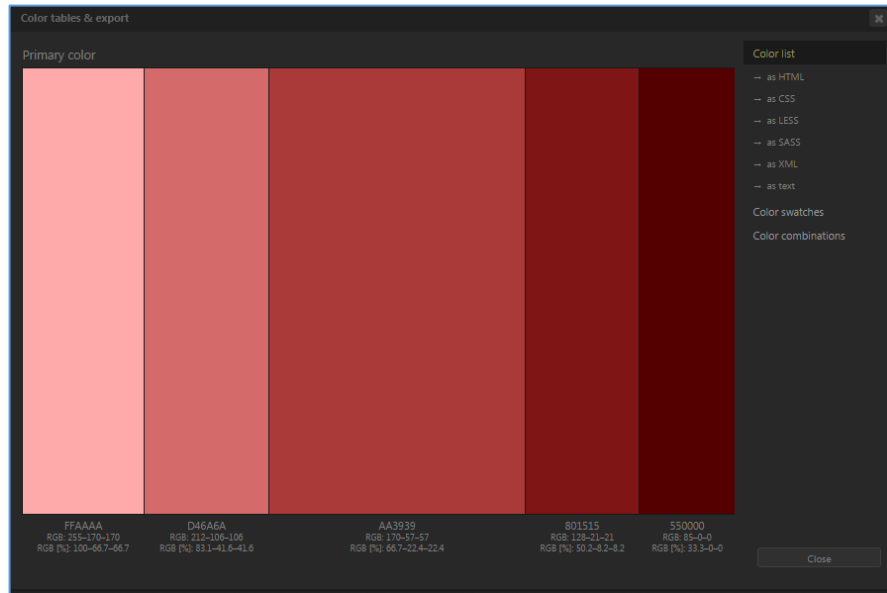
[Paletton](http://www.paletton.com)

<http://www.paletton.com>

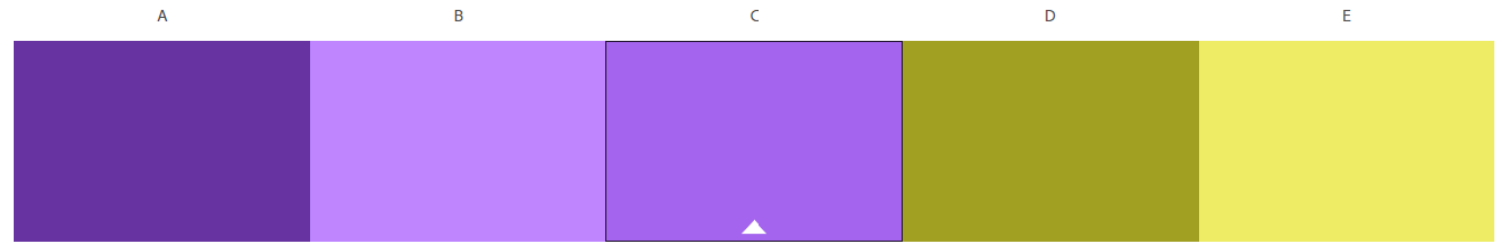


# Choosing Colors

One of the tasks in Phase 3 of the semester project is submitting the color palette you intend to use for your web site



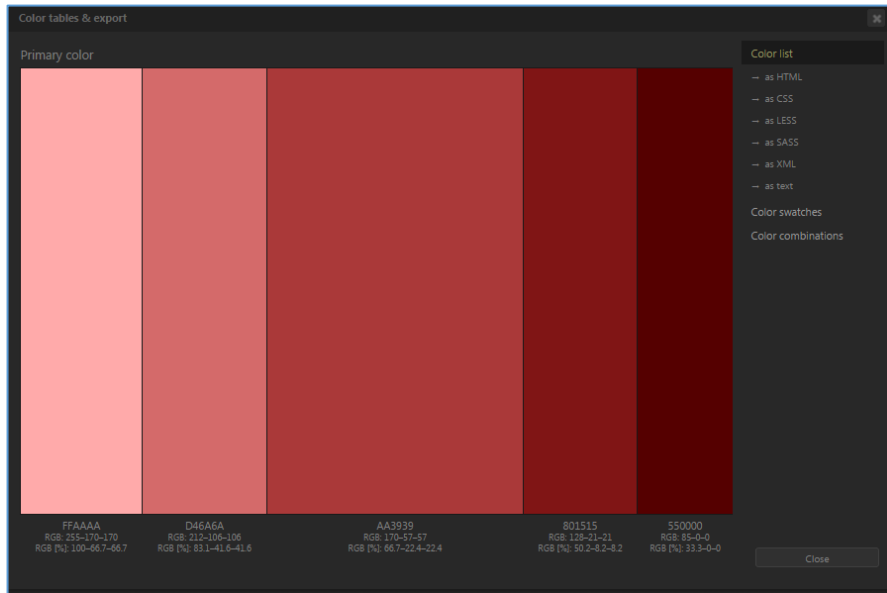
Paletton



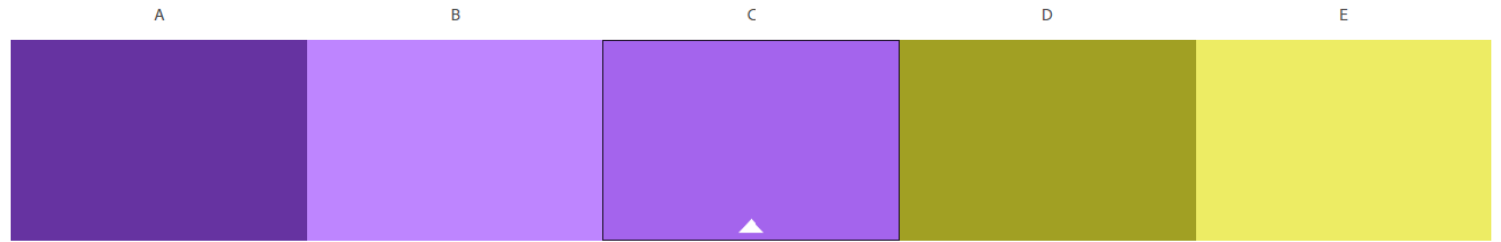
Adobe

# Choosing Colors

For this phase, **DO NOT** say, 'We intend to use shades of blue and gray,' or something like that. Pick a color palette, take a screen shot, and include it in your submission



Paletton



Adobe

# The Most Important Thing About Colors

The single most important issue with color selection is -- **Contrast**

When designing websites, it's important to make sure that you're creating sufficient color contrast

Color contrast is part of a broader set of web standards that fall under the umbrella of accessibility

They're meant to make sure that everyone – including those with visual, motor/mobility, auditory or cognitive impairments, or seizures – are not presented with barriers that prevent interaction with websites

# The Most Important Thing About Colors

Seriously, how easy is it to read this slide, compared to all of the others?

Even for folks who aren't visually impaired, low contrast can result in eye strain

This is part of what is referred to as a "negative user experience"

User experience (a positive one) is an over-arching goal of effective design

About 8% of all men and about 0.5% of all women are suffering from color blindness (color vision deficiency)

To put the percentages into perspective – there are close to 300 million people who are colorblind



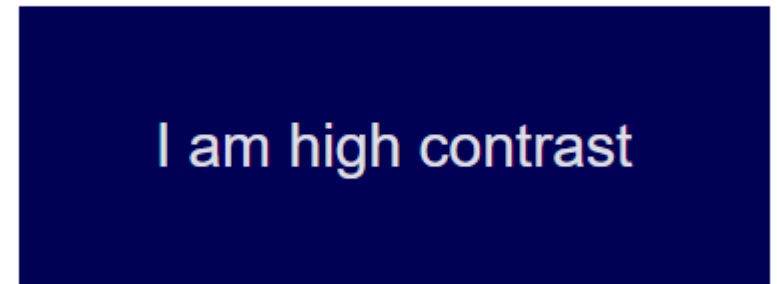
# The Most Important Thing About Colors

This is what we mean by contrast

Contrast is the difference between the foreground of an element and its background

The top image would be very difficult for someone with vision impairment to read

We'll talk about this later -- but accessibility to the web for everyone is a **HUGE** issue



# The Most Important Thing About Colors

Adobe Color and Paletton (and, incidentally, the web developer tools included in all modern browsers) will provide feedback regarding the contrast between the colors you've chosen

It is impossible to over-emphasize the importance of color selection

Many of us don't consider ourselves to be 'creative' in the sense of color selection -- these tools can help us overcome our shortcomings...



Donate



Triad (3-colors)

☐ add complementary

Hue: 0°

opposite

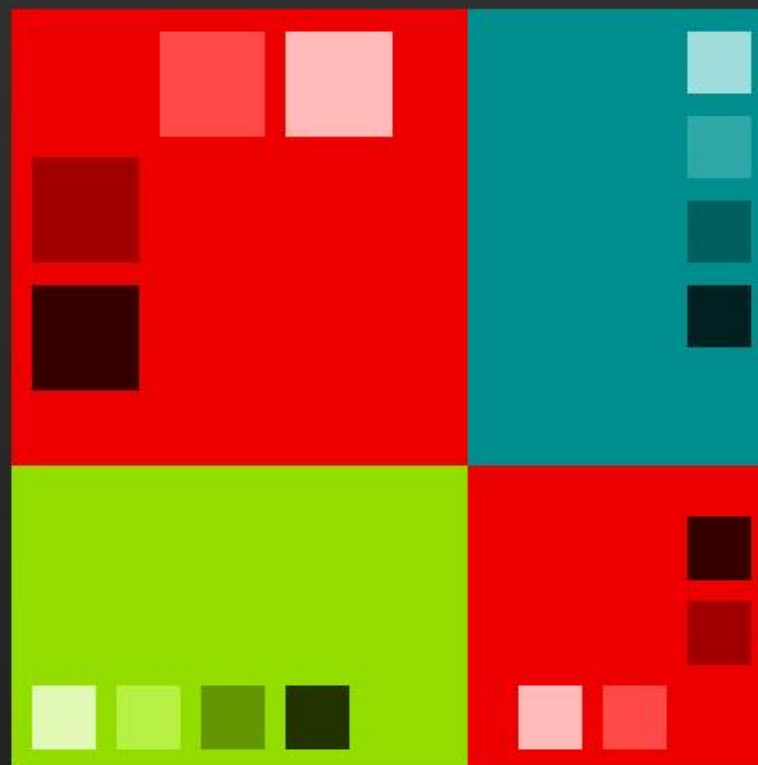
Dist: 30°

Base RGB:

ED0000

Fine Tune...

My Palette:



☐ Vision simulation ▾

COLORS

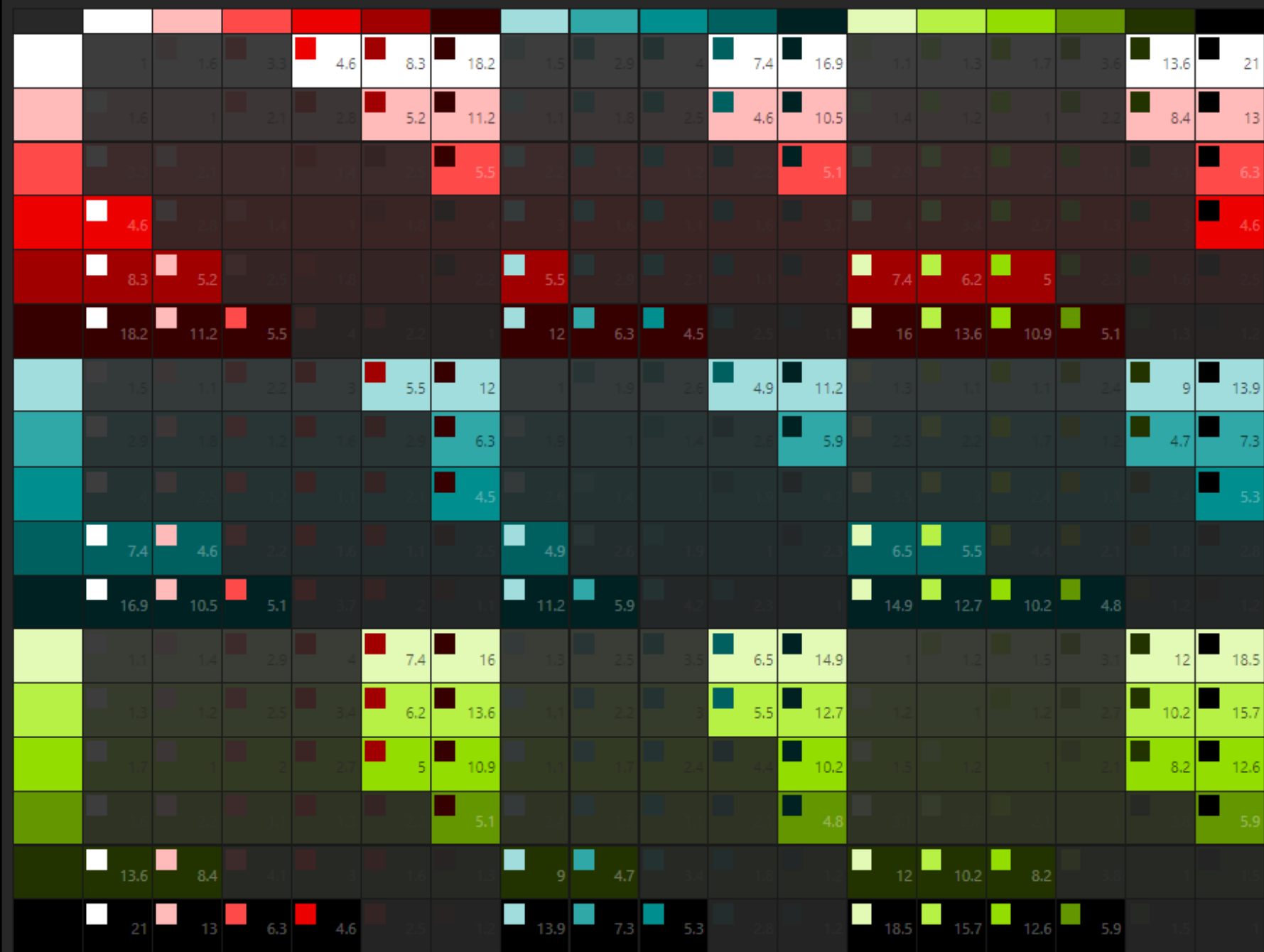
PRESETS

PREVIEW ▾

EXAMPLES...

TABLES / EXPORT...

Let's take a color sample  
swatch from  
paletton.com



- Color list
- Color swatches
- Color combinations

The minimum contrast accepted for accessibility is 4.5:1

Paletton.com will display the color combinations that meet this standard

The higher number, the higher contrast of combined colors.

Min. contrast: 4.5

Close

# Contrast

This, historically, has been a ticklish subject for our class

Again, a lot of us computing students don't think in these terms

But, web development includes a lot of folks with different educational backgrounds

It is important that we all have at least some understanding of what other folks' responsibilities are

That includes graphic design, digital media, marketing, computer science, sales, advertising, and a bunch of other stakeholders

**"Stakeholder"** == ANYONE who has an interest in the success of a web site



Kind of a  
big deal

# CSS - The Box Model

# The Box Model

You ever seen those Russian nesting (“Matroyska”) dolls?



The CSS Box Model is conceptually similar



# The Box Model

A **very important** concept in CSS

Applies to **block level** elements

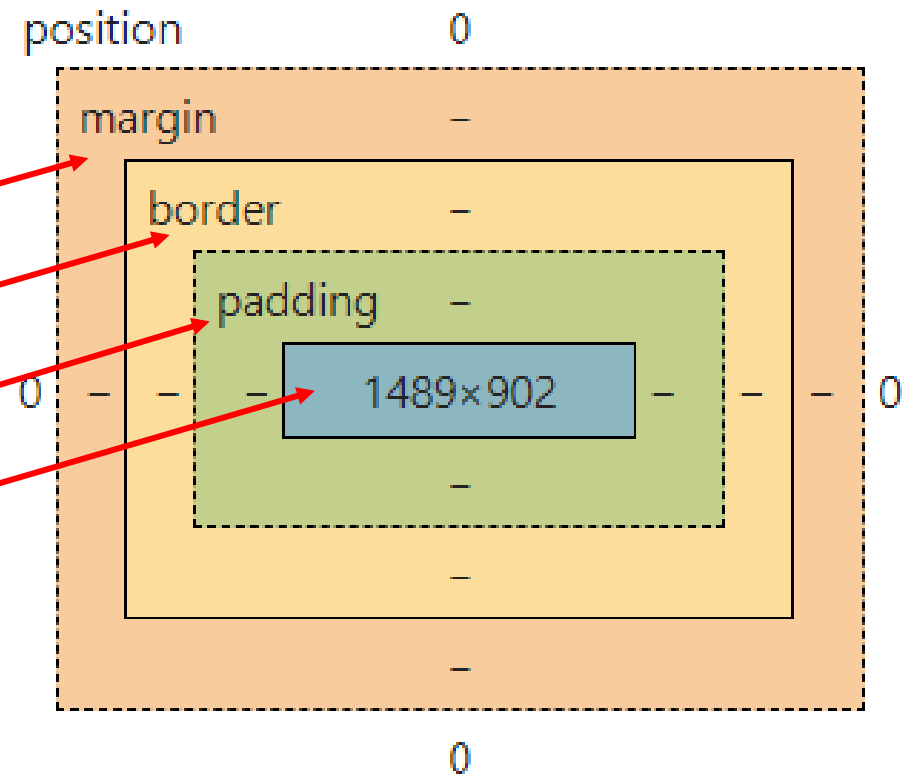
Each is regarded as a nested box with four parts:

Margin

Border

Padding

Content



# The Box Model

**Content:** The space where the element's content is displayed

**Padding:** Space between a box's border and content. Adding padding can improve readability of the content. Default = 0

**Border:** Wraps around the padding and content. Doesn't include the margin . Default = 0

**Margin:** Area outside of the border. Can create a gap between adjacent boxes . Default = 0

# The Box Model

By default, padding, border, and margin are 0

We can modify any or all of the three to modify the presentation of our content using CSS



# Box Dimensions

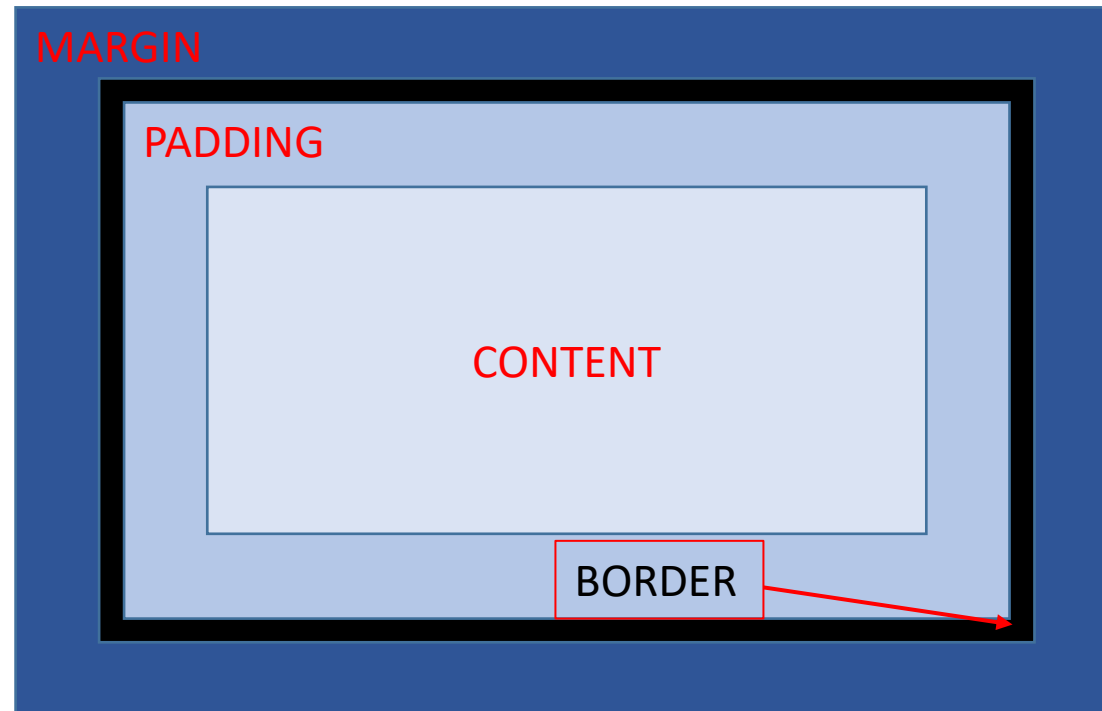
width, height

By default, a box is sized by a browser to be just big enough to hold its content

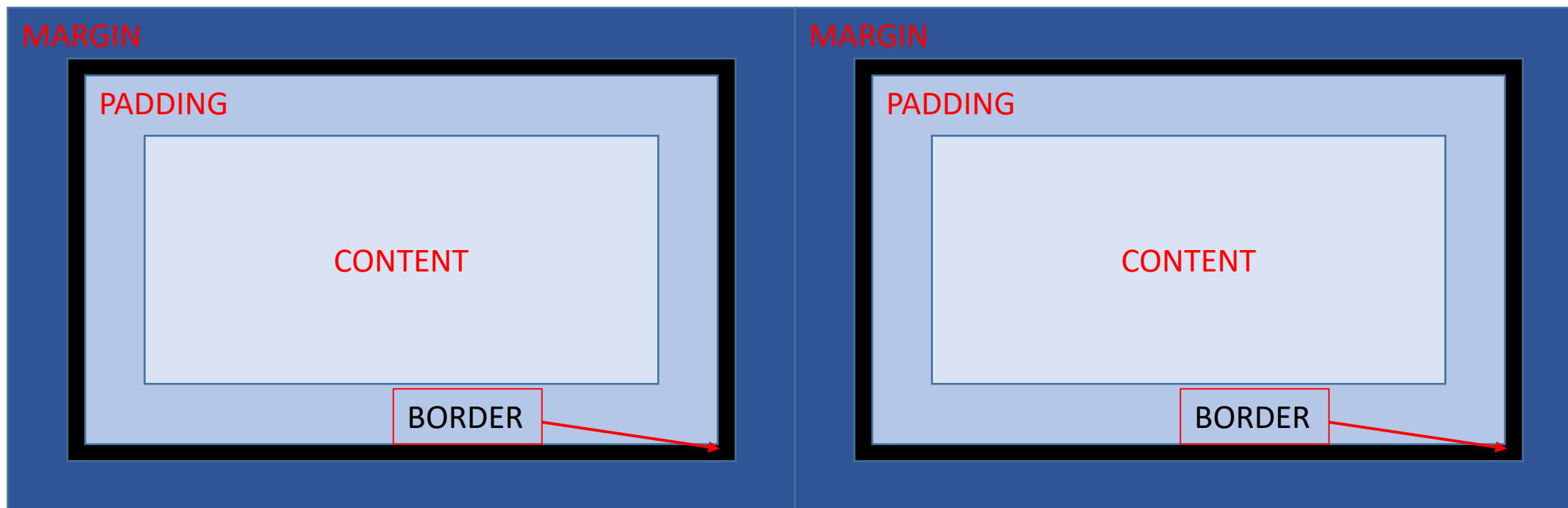
You can explicitly set the width and height in CSS

Height doesn't always work very well

# The Box Model



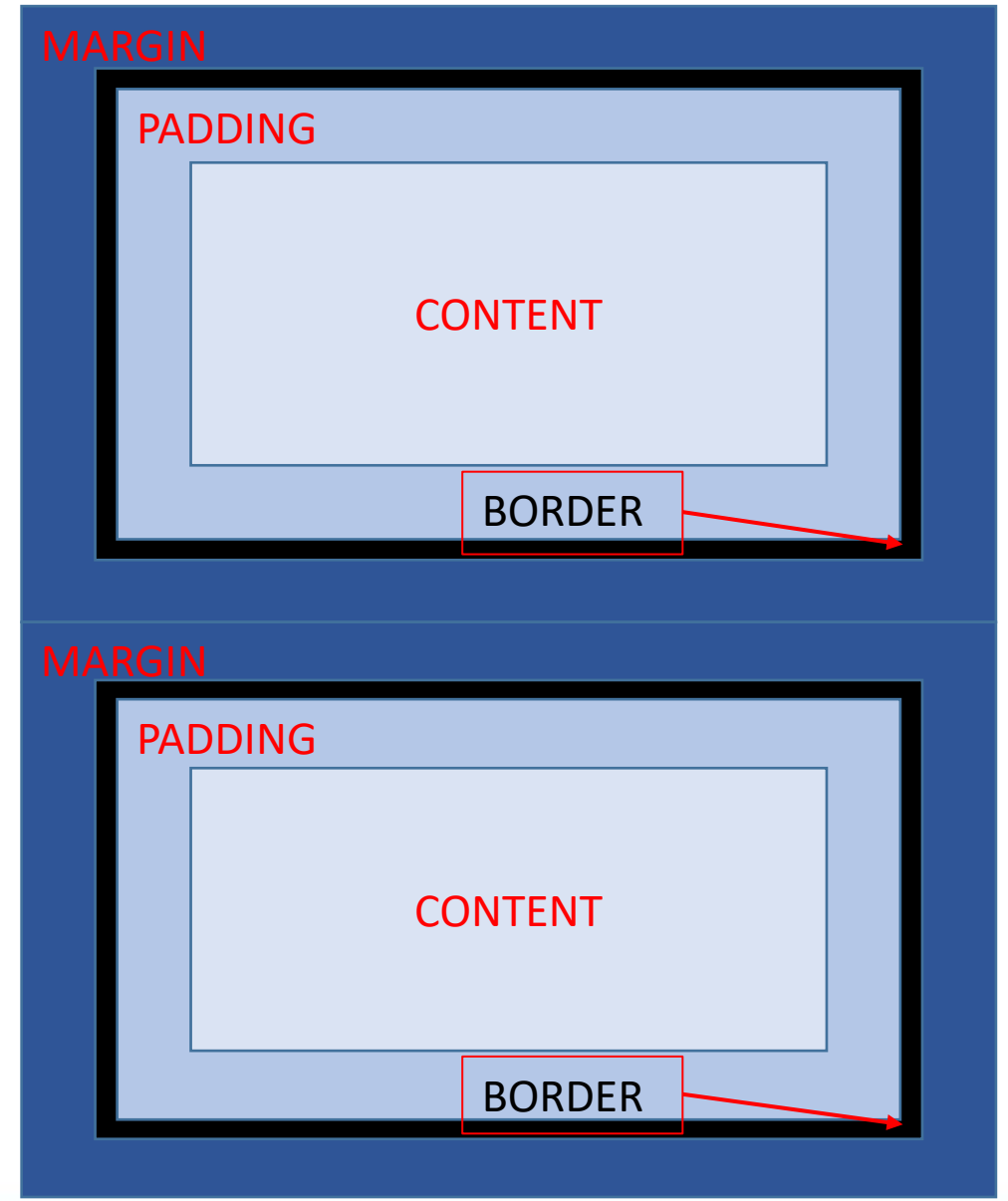
# The Box Model



# The Box Model - Margin Collapse

If two block-level elements are stacked one on top of the other, the distance will be whichever top or bottom margin is greatest

I.e., if the top box has a bottom margin of 25px and the bottom box has a top margin of 20px, the distance between them will be 25px



# The Box Model

So the box can be modified to visually separate one block of content from another, improving its presentation

Web design is about more than just writing code

Arranging content based on established best practices to make it attractive for visitors and thus likely to engage them in the future

Again, the concept falls under “User Experience,” or UX

# The Box Model

Padding, border, and margin can be set one of four ways:

Specifying top, right, bottom, and left values

Specifying top/bottom and right/left values

Specifying a single value that applies to top, right, bottom, and left

```
/* Syntax */
/* All four sides */
p {
    padding-top: 5px;
    padding-right: 5px;
    padding-bottom: 5px;
    padding-left: 5px;
}

/* top/bottom, right/left */
p {
    padding: 5px 5px;
}

/* all around */
p {
    padding: 5px;
}
```



# The Box Model

Padding, border, and margin can be set one of four ways:

Specifying all four  
on one line



The order of the values is top, right, bottom, left (clockwise, starting from the top)


```
/* all 4 values at once */  
p {  
  padding: 5px 5px 5px 5px;  
}
```

# A Note about Borders

To make a border work, you have to modify three properties:

border-width  
border-style  
border-color



```
p {  
    width:      100px;  
    height:     75px;  
    border-width: 1px;  
    border-style: solid;  
    border-color:  rgb(255,0,0);  
}
```




# A Note about Borders

We can (and often do) write this in shorthand

border



```
/* border shorthand */  
p {  
    width:      100px;  
    height:     75px;  
    border: 1px solid  rgb(255,0,0);  
}
```

# A Note about Borders

## Border styles

**dotted** - Defines a dotted border

**dashed** - Defines a dashed border

**solid** - Defines a solid border

**double** - Defines a double border

**groove** - Defines a 3D grooved border.  
The effect depends on the border-color value

**ridge** - Defines a 3D ridged border. The effect depends on the border-color value

**inset** - Defines a 3D inset border. The effect depends on the border-color value

**outset** - Defines a 3D outset border. The effect depends on the border-color value

**none** - Defines no border

**hidden** - Defines a hidden border

# A Note about Borders

**padding** and **margin** can be modified alone, without additional values

But **border** alone won't work unless all three properties are set (actually, you can leave off **border-color**; it defaults to black)

# Tables

# Tables

Tables have also been a part of the HTML language since Day 1

Historically, tables often used in HTML for page layout and text alignment. CSS techniques now used for this

**HTML tables should only be used for rendering data that belongs naturally in a grid**, in other words where the data describe a number of objects that have the same properties

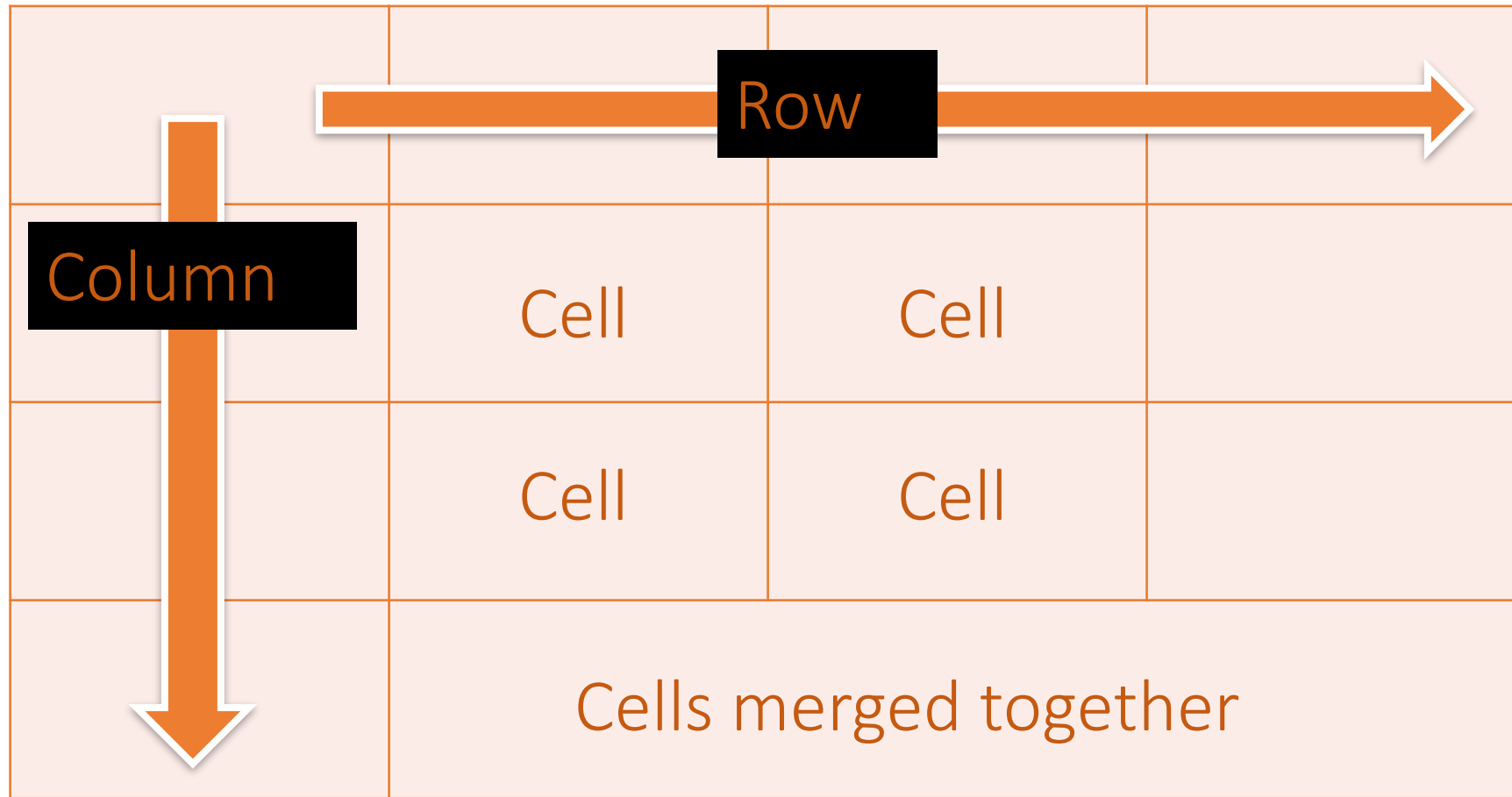
Tables should **never** be used for layout

# Tables

Order #	Item	Quantity	Total price
145	ZOTAC GeForce RTX 3070	1	\$500.00
146	AMD Ryzen 5 5600X CPU	1	\$280.00
147	ASRock X570 STEEL LEGEND MB	1	\$196.00
148	G.SKILL Aegis 16GB 3200MHz	2	\$79.00
149	Western Digital Blue 2TB	1	\$50.00

\* Note: I styled this example using CSS - We'll learn how to do that later

# Taxonomy of tables



# Table Tags

**<table></table>**

beginning and end of a table

**<tr></tr>** (table row)

beginning and end of a row

**<td></td>** (table data)

beginning and end of a cell

```
<!-- Constructing a table -->
<table>
  <tr>
    <td>Frodo</td><td>Samwise</td>
  </tr>
  <tr>
    <td>Merry</td><td>Pippin</td>
  </tr>
</table>
```

Tables are laid out row by row, top to bottom, left to right



# Table headings

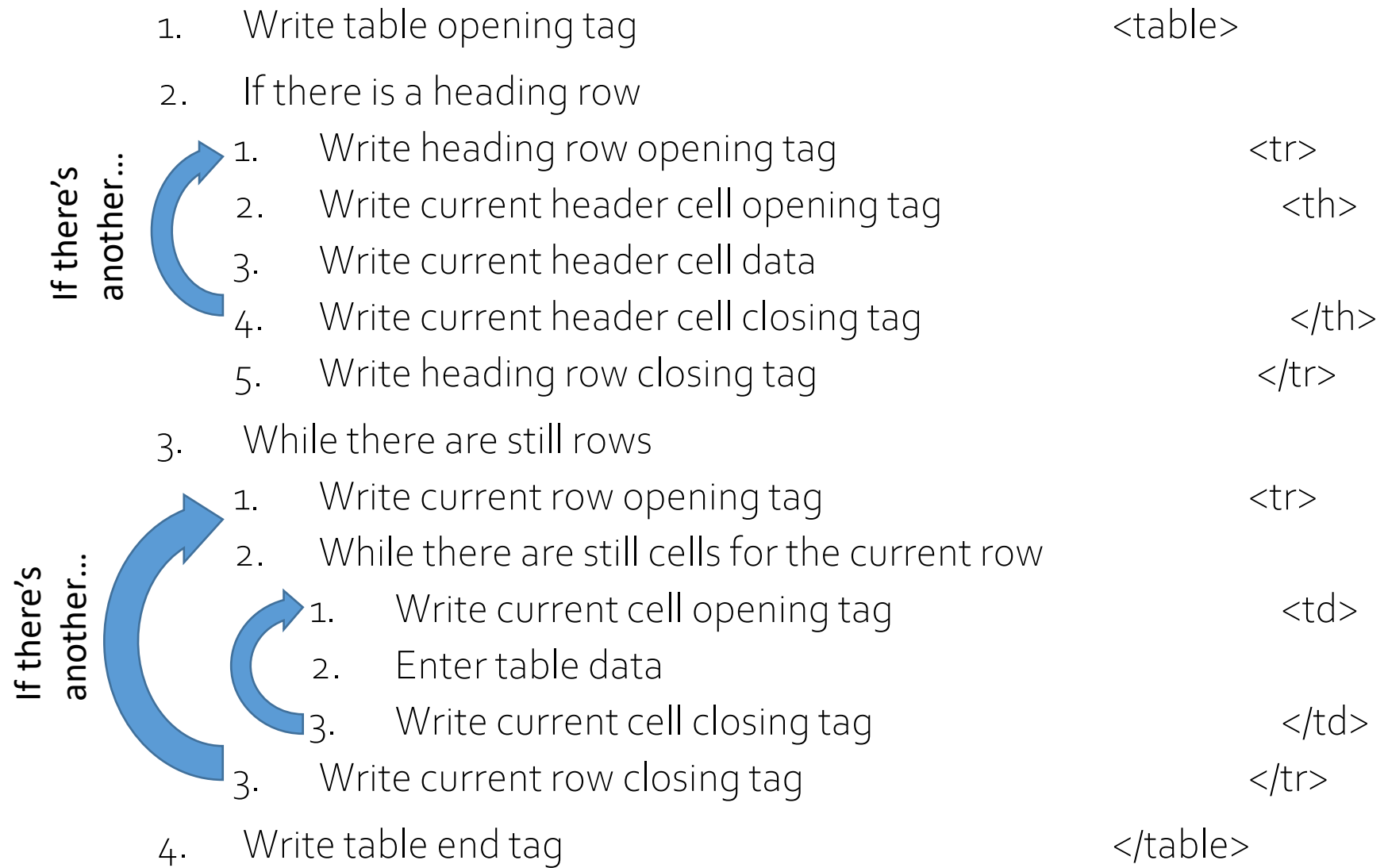
We often want to include column headings with our tables

**<th></th>** (table heading)  
beginning and end of a table heading

By default, all major browsers display  
**<th>** as **bold** and **center**

```
<!-- Constructing a table -->
<table>
  <tr>
    <th>Name</th><th>Name</th>
  </tr>
  <tr>
    <td>Frodo</td><td>Samwise</td>
  </tr>
  <tr>
    <td>Merry</td><td>Pippin</td>
  </tr>
</table>
```

# Tables, programmatically (an algorithm)



# Spanning Cells

Cells can be joined using **rowspan** (for vertical joining) and **colspan** (for horizontal joining) attribute on **td** tag

**rowspan**="n" or **colspan**="n"

**n** represents number of rows or columns to span across

```
<table>
  <tr>
    <td>1</td><td colspan="2">2</td>
  </tr>
  <tr>
    <td>3</td><td>4</td><td rowspan="2">5</td>
  </tr>
  <tr>
    <td>6</td><td>7</td>
  </tr>
</table>
```

# Spanning Cells

1	2	
3	4	5
6	7	

# Spanning Cells

When writing code for a table that includes merged cells, pencil and paper can make the job easier

Remember, HTML draws tables row by row

Draw out the proposed table by hand

It may help to put `<tr>` tags around each row and `<td>` tags above each column

# Spanning Cells

	<td>	<td>	<td>	<td>	
<tr>	<td>	<td colspan="3">			</tr>
<tr>	<td rowspan="3">	<td>	<td>	<td>	</tr>
<tr>		<td rowspan="2">	<td>	<td>	</tr>
<tr>			<td colspan="2">		</tr>

Later (not in this class), when you're writing code that "writes the code" this can help. Think: Ebay  
-- search results are displayed in dynamically generated tables

# Spanning Cells

<tr>	<td>	<td>	<td>	<td>
	1	2		
<tr>	3	4	5	6
		7	8	9
<tr>			10	
<tr>				

```
<table>
  <tr>
    <td>1</td><td colspan="3">2</td>
  </tr>
  <tr>
    <td rowspan="3">3</td><td>4</td><td>5</td><td>6</td>
  </tr>
  <tr>
    <td rowspan="2">7</td><td>8</td><td>9</td>
  </tr>
  <tr>
    <td colspan="2">10</td>
  </tr>
</table>
```



# Spanning Cells

So, using our hand drawn diagram, the code for a complex table can easily be written

Remember, the number of cells in each row must be equal

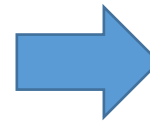
Cells that span rows or columns have to be accounted for



# Tables – Fixing Column Width

We can fix the width of tables' columns by using standalone tags inside the table element in the HTML with inline CSS

```
<table>
  <col style="width:15%;">
  <col style="width:55%;">
  <col style="width:30%;">
  <tr><td>1</td><td>2</td><td>3</td></tr>
  <tr><td>4</td><td>5</td><td>6</td></tr>
</table>
```



1	2	3
4	5	6

# Dynamic Table-Building

# Dynamic Tables

This is outside the scope of this class, but I want to illustrate how tables can be built dynamically

Again, think Ebay - you enter a search term, the application fetches the results from a database, and the results are displayed in a table

The table doesn't exist until the results are back

# Dynamic Tables

We'll see how this table is built (using JavaScript) when the user clicks the button

Show Table



	CPU	ZOTAC GeForce RTX 3070
	1	\$500.00
	CPU-Cooler	EK EK-AIO 240 D-RGB 66.04 CFM Liquid CPU Cooler
	1	\$179.99
	Motherboard	Asus TUF GAMING X570-PLUS ATX AM4 Motherboard
	1	\$192.09
	RAM	OLOy WarHawk RGB 32 GB (2 x 16 GB) DDR4-3600 CL16 Memory
	2	\$313.98
	Storage	ADATA XPG SX6000 Pro 1 TB M.2-2280 NVME Solid State Drive
	2	\$91.99
	Graphics Card	EVGA GeForce RTX 3090 24 GB XC3 ULTRA GAMING Video Card
	1	\$2555.49

```
<body>
  <div id="tab"></div>
  <button id="b">Show Table</button>

  <script>
    let list = [
      {
        "image": "images/cpu.jpg",
        "item": "CPU",
        "description": "ZOTAC GeForce RTX 3070",
        "quantity": "1",
        "price": "$500.00"
      },
      {
        "image": "images/cpu-cooler.jpg",
        "item": "CPU-Cooler",
        "description": "EK EK-AIO 240 D-RGB 66.04 CFM Liquid CPU Cooler",
        "quantity": "1",
        "price": "$179.99"
      },
    ],
```

<div> starts out empty

Button

List of items  
(retrieved, IRL,  
from a  
database)

These are keys.  
Notice how they  
repeat (with  
different values)  
for each item

```
let out = "<table>";
for (item in list) {
  let el = list[item];
  out += "<tr>" +
    "<td rowspan='2'><img src='" + el.image + "'>" +
    "</td><td>" + el.item + "</td><td>" +
    el.description + "</td>" +
    "</tr>" +
    "<tr>" +
    "<td>" + el.quantity + "</td>" +
    "<td colspan='2'>" + el.price + "</td>" +
    "</tr>";
}
out += "</table>";

document.getElementById("b").addEventListener("click", function() {
  document.getElementById("tab").innerHTML = out;
});
```

Output string (this will hold the HTML for the table)

For each item in the list

Build each pair of rows and add them to the table

When the button is clicked,

Add the code to the page

# Dynamic Tables

We'll see how this table is built (using JavaScript) when the user clicks the button

Show Table



	CPU	ZOTAC GeForce RTX 3070
	1	\$500.00
	CPU-Cooler	EK EK-AIO 240 D-RGB 66.04 CFM Liquid CPU Cooler
	1	\$179.99
	Motherboard	Asus TUF GAMING X570-PLUS ATX AM4 Motherboard
	1	\$192.09
	RAM	OLOy WarHawk RGB 32 GB (2 x 16 GB) DDR4-3600 CL16 Memory
	2	\$313.98
	Storage	ADATA XPG SX6000 Pro 1 TB M.2-2280 NVME Solid State Drive
	2	\$91.99
	Graphics Card	EVGA GeForce RTX 3090 24 GB XC3 ULTRA GAMING Video Card
	1	\$2555.49



# Dynamic Tables

Again, don't feel intimidated if the last two slides read like Greek to you

The point is, a lot of what we'll be learning this semester will form the basis for future classes (like CSCI 1720, CSCI 2910, and CSCI 3110)

A little preview can't hurt



# Styling Tables

# Styling Tables

The basic, un-styled table, can be awful un-inspiring

Cells automatically adjust to the content of the widest cell in the column

No borders	<b>Order #</b>	<b>Item</b>	<b>Quantity</b>	<b>Total price</b>
No padding	145	ZOTAC GeForce RTX 3070	1	\$500.00
	146	AMD Ryzen 5 5600X CPU	1	\$280.00
	147	ASRock X570 STEEL LEGEND MB	1	\$196.00
	148	G.SKILL Aegis 16GB 3200MHz	2	\$79.00
	149	Western Digital Blue 2TB	1	\$50.00

# Styling Tables


We can add styling rules to tables to make them more attractive

It's a little more complicated, though, because of the fact that there are four elements - table, table row, table heading, and table data that comprise a table

There are actually some other table elements, but these are the ones we're focusing on

# Styling Tables

Let's start with the table element

```
table {  
  border: 1px solid  #777;  
  width: 600px;  
  margin: 50px auto;  
}
```

Order #	Item	Quantity	Total price
145	ZOTAC GeForce RTX 3070	1	\$500.00
146	AMD Ryzen 5 5600X CPU	1	\$280.00
147	ASRock X570 STEEL LEGEND MB	1	\$196.00
148	G.SKILL Aegis 16GB 3200MHz	2	\$79.00
149	Western Digital Blue 2TB	1	\$50.00

So, we added a border. Notice, though, that the border only goes around the table, not the cells

Also, we set a width, which spaces things out more nicely

Notice the margin. When we set the left and right margins on a block element to 'auto,' the element will be centered in its parent container (you can't really tell from the screenshot)

# Styling Tables

If we want borders around the cells, also:

```
th, td {  
    border: 1px solid #777;  
}
```

Order #	Item	Quantity	Total price
145	ZOTAC GeForce RTX 3070	1	\$500.00
146	AMD Ryzen 5 5600X CPU	1	\$280.00
147	ASRock X570 STEEL LEGEND MB	1	\$196.00
148	G.SKILL Aegis 16GB 3200MHz	2	\$79.00
149	Western Digital Blue 2TB	1	\$50.00

That's a little better

But that “double-border” is kind of ugly


Let's fix that

BTW, using the comma (`,`), we can apply the same rules to multiple selectors. We'll talk more about that later



# Styling Tables

Let's fix the borders:

```
table {  
  border: 1px solid  #777;  
  width: 600px;  
  margin: 50px auto;  
  border-collapse: collapse;  
  font-family: sans-serif;  
}
```

Order #	Item	Quantity	Total price
145	ZOTAC GeForce RTX 3070	1	\$500.00
146	AMD Ryzen 5 5600X CPU	1	\$280.00
147	ASRock X570 STEEL LEGEND MB	1	\$196.00
148	G.SKILL Aegis 16GB 3200MHz	2	\$79.00
149	Western Digital Blue 2TB	1	\$50.00

The `border-collapse` property applies to tables and, well, makes the borders collapse

We also changed the font-family to 'sans-serif,' well, because we kind of hate serif fonts



# Styling Tables

Now, it looks a lot better, but things are still kind of “scrunched up.” Let’s fix that:

```
th, td {  
    border: 1px solid #777;  
    padding: 5px;  
}
```

Order #	Item	Quantity	Total price
145	ZOTAC GeForce RTX 3070	1	\$500.00
146	AMD Ryzen 5 5600X CPU	1	\$280.00
147	ASRock X570 STEEL LEGEND MB	1	\$196.00
148	G.SKILL Aegis 16GB 3200MHz	2	\$79.00
149	Western Digital Blue 2TB	1	\$50.00

This is what the `padding` property does - remember: it adds space between the content and the border

# Styling Tables

How about this?

```
th, td {  
  border: 1px solid #777;  
  padding: 5px;  
}  
  
th {  
  background-color: #000055;  
  color: #ffffff;  
}
```

Order #	Item	Quantity	Total price
145	ZOTAC GeForce RTX 3070	1	\$500.00
146	AMD Ryzen 5 5600X CPU	1	\$280.00
147	ASRock X570 STEEL LEGEND MB	1	\$196.00
148	G.SKILL Aegis 16GB 3200MHz	2	\$79.00
149	Western Digital Blue 2TB	1	\$50.00

This illustrates the 'cascading' part of Cascading Style Sheets

We added rules that apply to both the `<th>` & `<td>` elements, then a more specific rule that applies only to `<th>` elements

# Styling Tables

Now, what we've created so far still doesn't look like the examples from earlier

There's a lot more fun stuff we can do to style tables

But, I guess, that's enough for today 😊

# Styling Tables

How do we get from here to there?

Order #	Item	Quantity	Total price
145	ZOTAC GeForce RTX 3070	1	\$500.00
146	AMD Ryzen 5 5600X CPU	1	\$280.00
147	ASRock X570 STEEL LEGEND MB	1	\$196.00
148	G.SKILL Aegis 16GB 3200MHz	2	\$79.00
149	Western Digital Blue 2TB	1	\$50.00

Order #	Item	Quantity	Total price
145	ZOTAC GeForce RTX 3070	1	\$500.00
146	AMD Ryzen 5 5600X CPU	1	\$280.00
147	ASRock X570 STEEL LEGEND MB	1	\$196.00
148	G.SKILL Aegis 16GB 3200MHz	2	\$79.00
149	Western Digital Blue 2TB	1	\$50.00

...time will tell...

# Questions?



# Lecture Quiz

1. What is the most important issue with regard to color selection?
  - A. Contrast
  - B. What tools you use
  - C. Hue
  - D. Saturation



# Lecture Quiz

2. What is contrast?

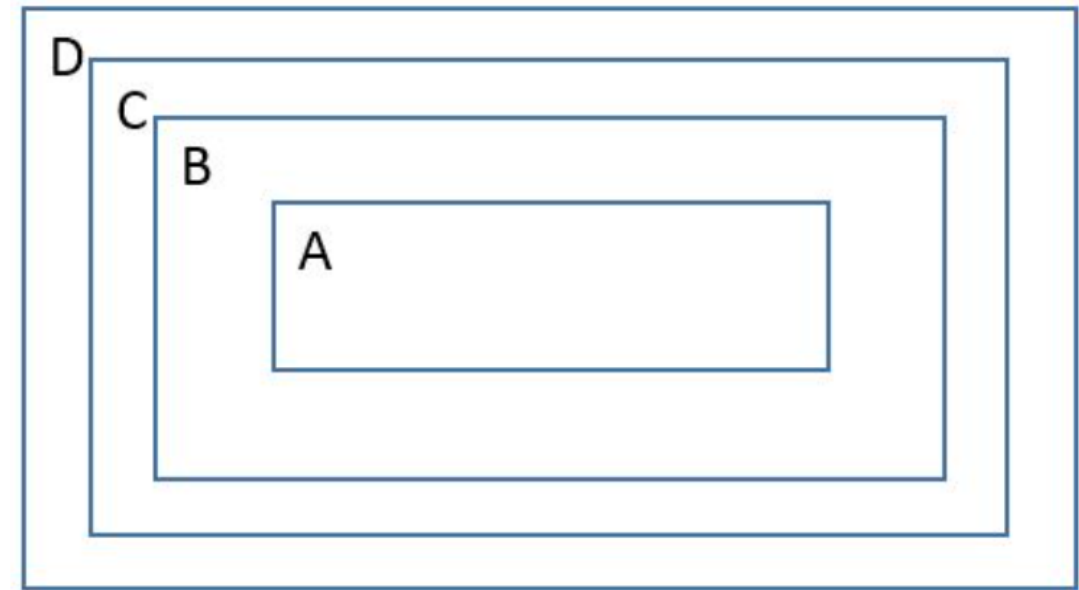
- A. A color's hue
- B. A color's saturation
- C. How pleasing a color may be for a given user
- D. The difference between foreground and background colors



# Lecture Quiz

3. CSS uses the Box Model when calculating the box properties. In the picture below, what does 'D' represent??

- A. Content
- B. Padding
- C. Border
- D. Margin



# Lecture Quiz

4. The box model can be applied to both block and inline elements

A. True

B. False

# Lecture Quiz

5. Tables are a relatively new element in HTML

A. True

B. False

# Lecture Quiz

6. Which of the following is used to create a cell in a table?

- A. `<table>`
- B. `<tr>`
- C. `<td>`
- D. `<tbody>`

# Lecture Quiz


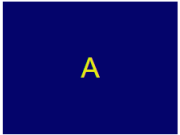
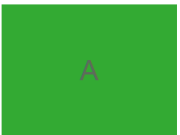
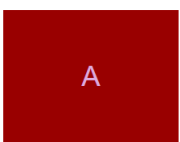
7. Which attribute is used to create cell #5 in this table?

- A. spancol
- B. rowspan
- C. colspan
- D. spanrow

1	2	
3	4	5
6	7	

# Lecture Quiz

8. Which of the following represents high contrast?

- A. 
- B. 
- C. 
- D. 

# Lecture Quiz

9. If I don't want my table to have double borders, how can I fix it?

- A. `border-collapse: collapse;`
- B. `collapse-border: true;`
- C. `border-collapse: true;`
- D. `collapse-border: collapse;`



# Lecture Quiz

10. By default, the <th> element displays text bold and centered

A. True

B. False

# Sources

- "HTML Reference", W3Schools, Retrieved from <http://www.w3schools.com/tags/default.asp>
- "HTML Tables", W3Schools, Retrieved from [http://www.w3schools.com/html/html\\_tables.asp](http://www.w3schools.com/html/html_tables.asp)
- "CSS Reference", W3Schools, Retrieved from <http://www.w3schools.com/cssref/default.asp>
- "HTML Tables – when and how to use tables in HTML", Ben Hunt's Web Design from Scratch, Retrieved from <http://webdesignfromscratch.com/html-css/html-tables/>

# Copyrights



Presentation prepared by and copyright of John Ramsey,  
East Tennessee State University, Department of  
Computing . ([ramseyjw@etsu.edu](mailto:ramseyjw@etsu.edu))



- Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.
- IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.
- Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.
- Oracle is a registered trademark of Oracle Corporation.
- HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.
- Java is a registered trademark of Sun Microsystems, Inc.
- JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.
- SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.
- Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.
- ERPsims is a registered copyright of ERPsims Labs, HEC Montreal.
- Other products mentioned in this presentation are trademarks of their respective owners.