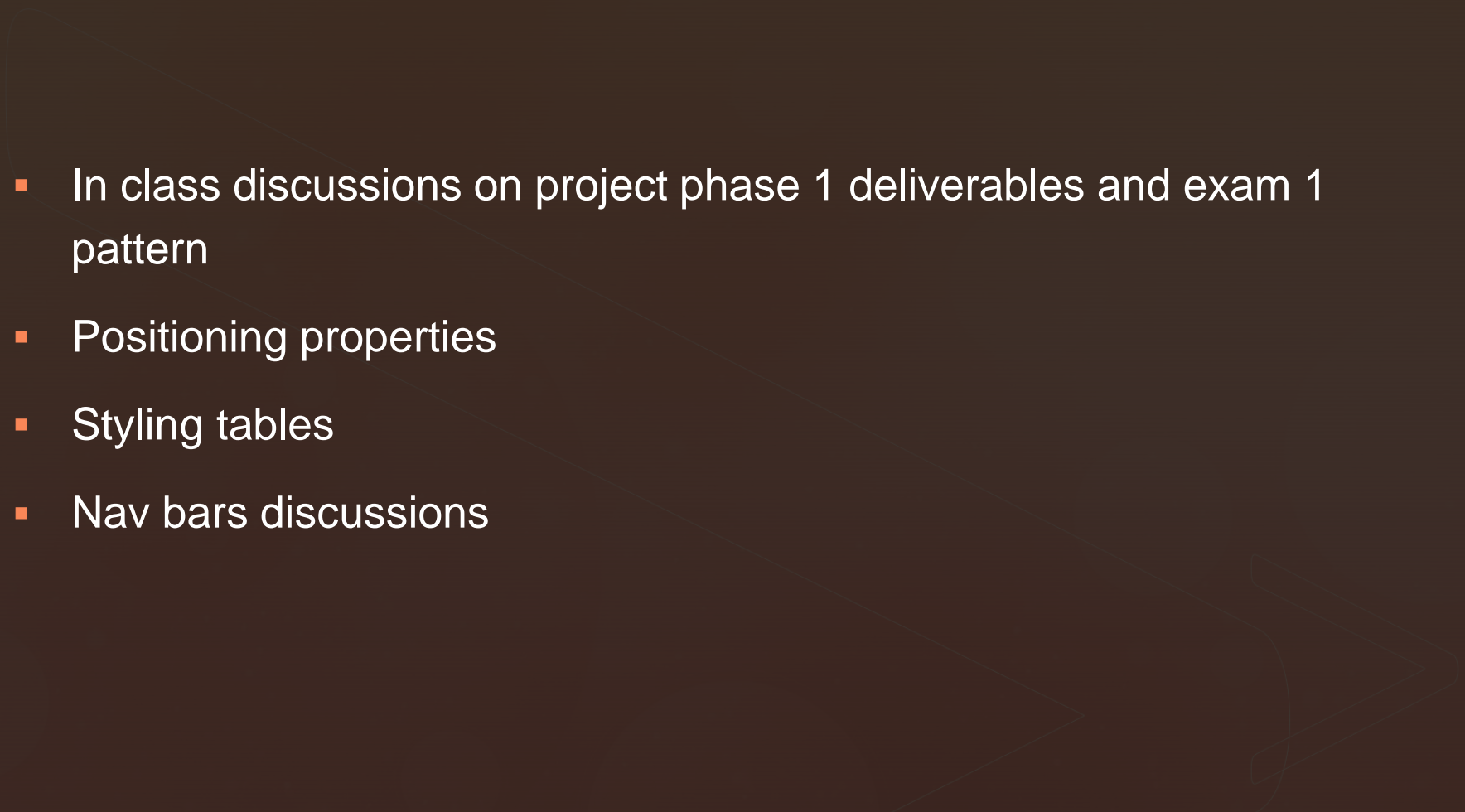




# CSS part 4



# Agenda

- In class discussions on project phase 1 deliverables and exam 1 pattern
  - Positioning properties
  - Styling tables
  - Nav bars discussions
- 



# Positioning Properties

# Positioning and Page Layout

- Positioning helps determining the page layout – where and how your elements will be positioned on the page
- It has four properties
  - Static
  - Relative
  - Absolute
  - Fixed
- Element positions can be further modified using properties such as
  - Top, right, bottom, left.
- For example: you may choose element to be **relative** and 30px from the **top**

# Static positioning

- Static is the default positioning property.
- If you don't specify any positioning, then static is used.
- In static positioning, browser places the elements in the next available space/position
- Specifying top, right, bottom, left properties doesn't affect elements with static properties

# Relative Positioning

- Change the default way in which browser positions elements
- In relative properties, elements can be positioned relative to itself
  - Its like using static positioning but being able to add some offsets
- Note that – relative positioning does not affect neighboring elements.
  - If you offset it too much – the element might just vanish without repositioning neighboring elements
- How to use – relative position are used generally for *container* elements



# Absolute Positioning

- If you state that an element has ***absolute*** positioning the following things will happen
  - The element will no longer be considered as a part of the normal page layout and positioning
  - The absolute element is will be positioning relative to its nearest parent node or the root node (in DOM), whichever is there.
  - Other elements will no longer consider the element to be present
- All of these may lead to overlapping elements

# Fixed Positioning

- If an element is with ***fixed*** positioning, it is placed relative to the browser window
  - E.g. think of those pop-up windows that stays in that one location on the page regardless where you scroll
- If you want to make the nav bar appear *always on top* you give it fixed positioning.



# Stacking elements: Z-index

- If you do end up creating a design that requires elements to be placed on top of each other, and
- You want to control the order in which the stacking is done
- Use the Z-index
  - A numeric (positive or negative) value that controls the order of stacking
  - Value spectrum - +100 always on top and -100 always on bottom



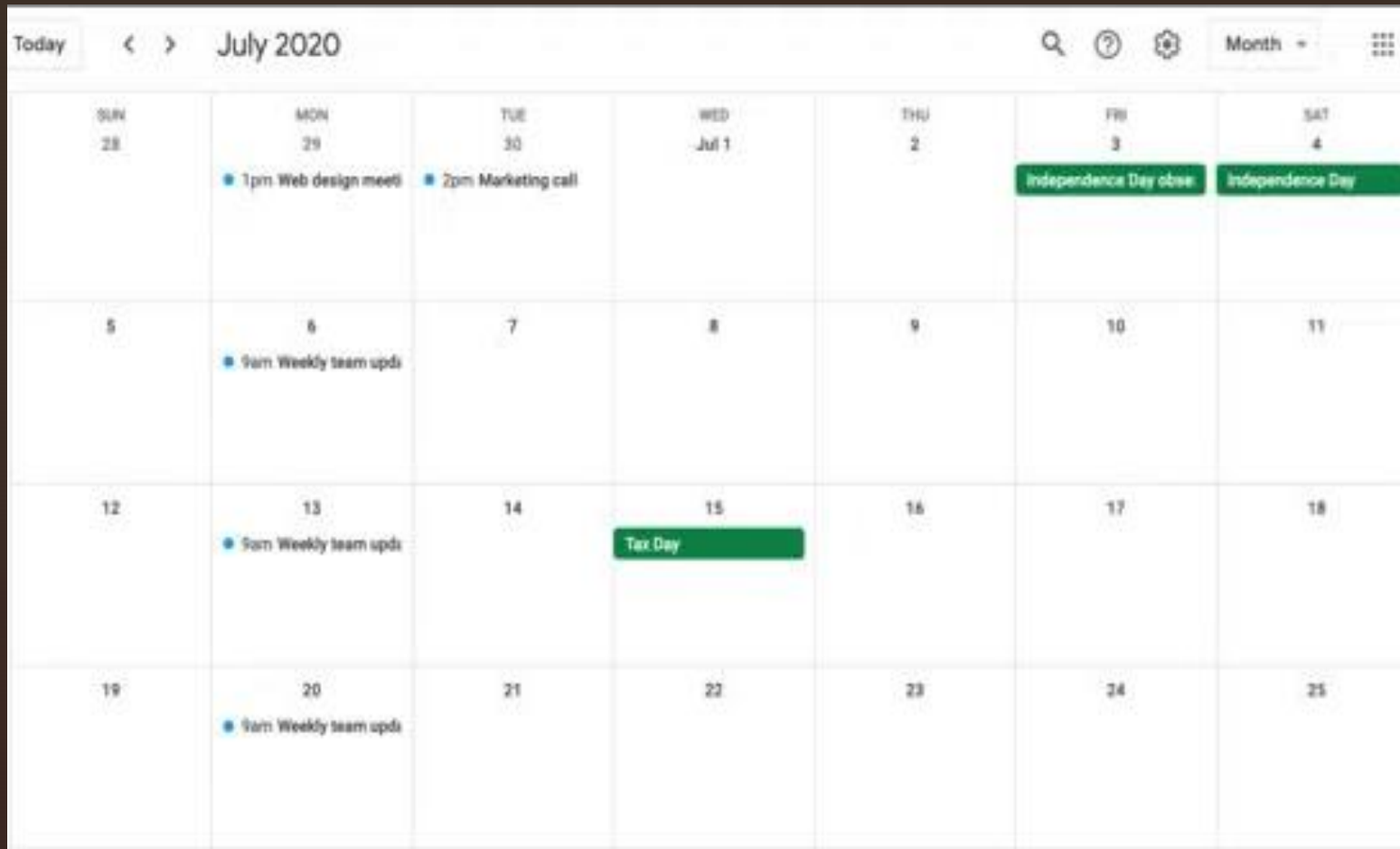
# Styling HTML Tables



# Styling Tables

- Tabular information is not limited to showing scientific data
- You can use tables for more generic purposes, that might require some styling rules
  - A calendar
- When it comes to styling tables, a calendar is an example that will really make you think how you want to do it.

# styling considerations for calendars



Can you color the alternating rows and columns?

Can you color the special events and holidays?

Can you hide the events and show it only when users interact with it?

Can you change text alignments/colors etc?

Can you develop something like this from scratch using HTML and CSS only?



# Navigation Menus



# Nav Bars

- The two basic styling for nav bars are
  - Vertical
  - Horizontal
- In class discussions