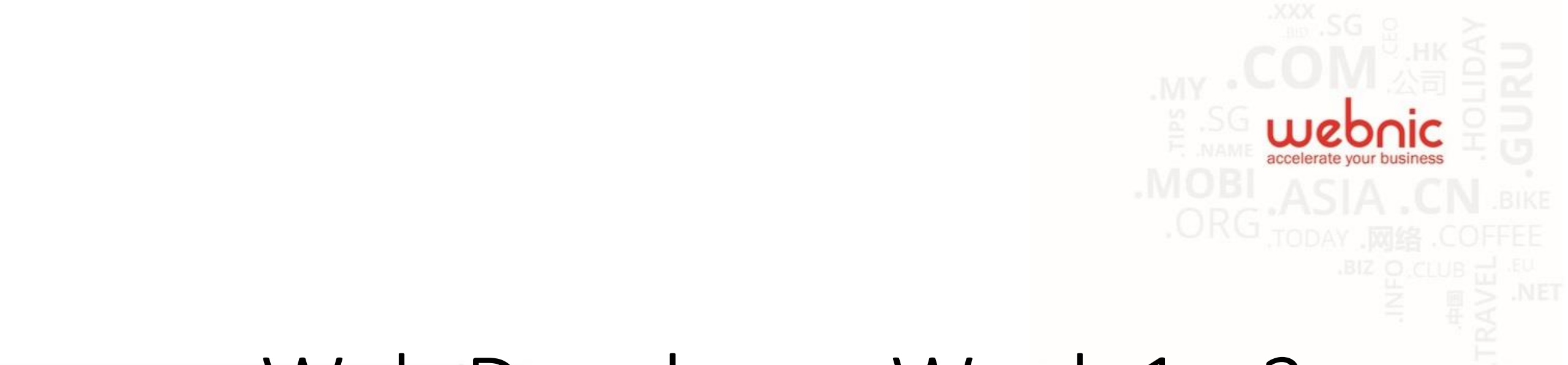


Web Developer Week 1 - 2

Advance HTML & CSS

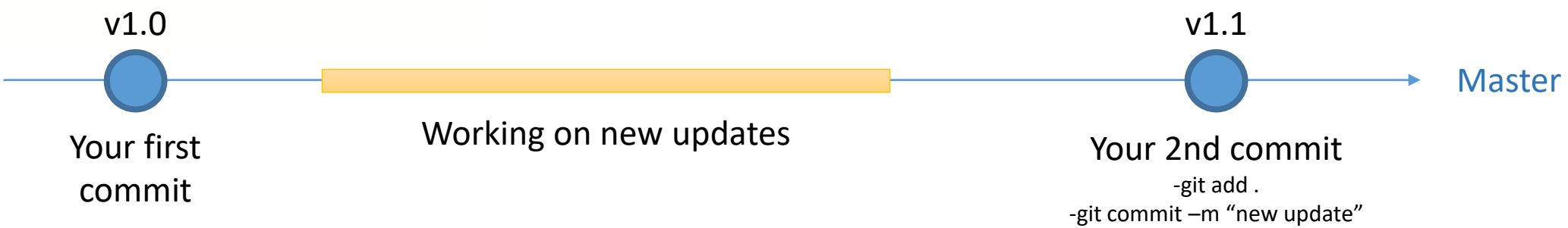
Recap…

- Give example of followings:
  - Element
  - Attributes
  - Entities
- What are different ways to include CSS into your HTML file?
- Cited the CSS properties for the following:
  - Aligning text center
  - Underline a text
  - Italic a text
  - Bold a text
- How to push your code to github

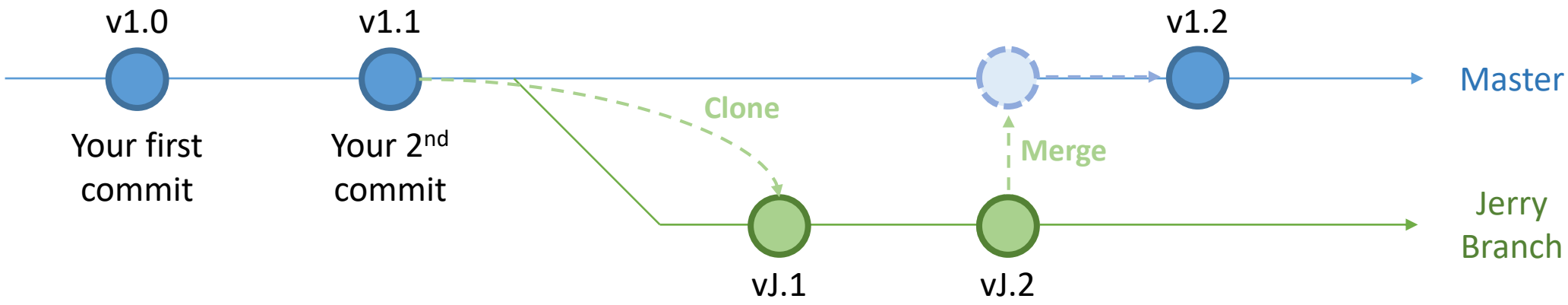
Recap…

- How to push your code to github for the first time
  - Git init
    - Initialize git
  - Git remote add origin <link>
    - Add a repository link for this project
  - Git add .
    - Select all files within this folder
  - Git commit –m “message”
    - “I confirm want to use” all the selected files
  - Git push origin master
    - Upload the file from your local to the online repository, and merge it.
- How to push your code after changes
  - Git add .
  - Git commit –m “message”
  - Git push origin master

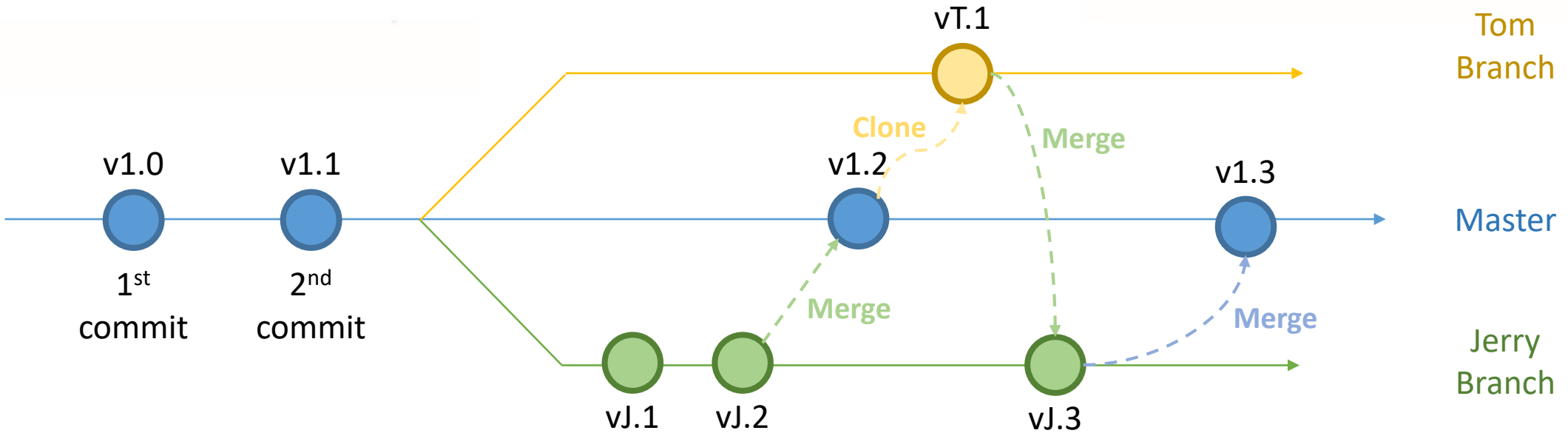
What is Git


v1.0
v1.1
Master
Your first commit
Working on new updates
Your 2nd commit
-git add .
-git commit –m “new update”

What is Git

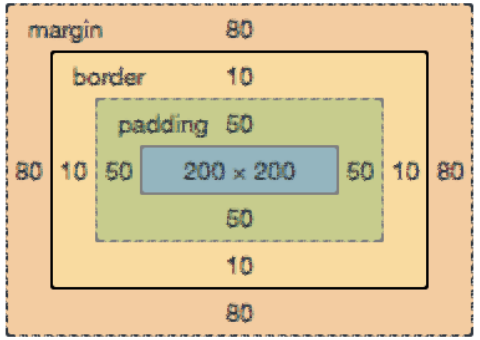

v1.0
v1.1
v1.2
Master
Your first commit
Your 2nd commit
Clone
Merge
Jerry Branch
vJ.1
vJ.2

What is Git


vT.1
Tom
Branch
Clone
Merge
v1.0
v1.1
v1.2
v1.3
Master
1st
commit
2nd
commit
Merge
Merge
Jerry
Branch
vJ.1
vJ.2
vJ.3

CSS Box Model

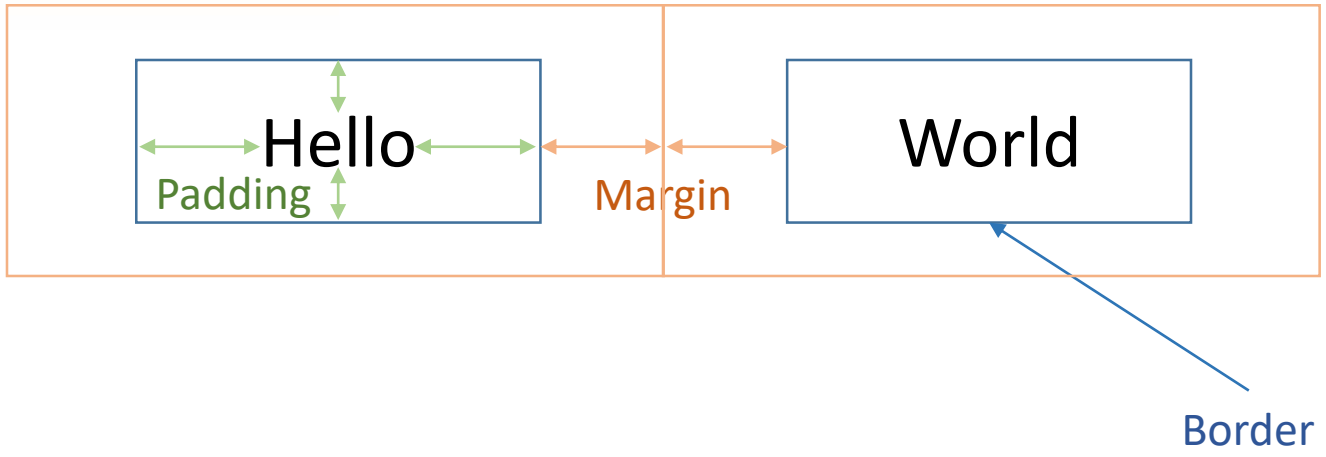
- Every block element within a document is structured as a rectangular box inside the document layout, the size and "onion layers" of which can be tweaked using some specific CSS properties. The relevant properties are as follows:
  - Width & height
  - Padding
  - Border
  - Margin



Margin vs Padding

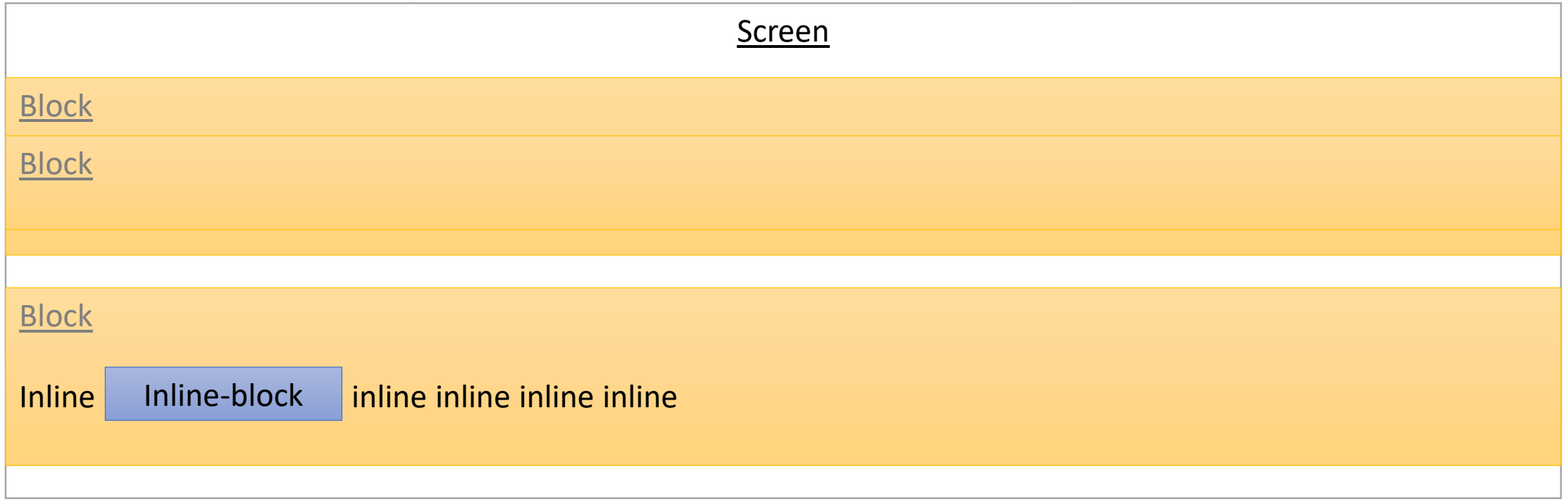
- Margin
  - The margin represents the outer area surrounding the CSS box
  - You can specify values for each side of box using:
    - margin-top
    - margin-right
    - margin-bottom
    - margin-left
  - You can also use a shorthand
    - Margin: top right bottom left
    - Margin: 5px 10px 5px 20px
- Padding
  - The padding property allows us to specify how much space should appear between the content of an element and its border
  - You can specify the value of a box using:
    - padding-top
    - Padding-right
    - Padding-bottom
    - Padding-left
  - You can also use a shorthand
    - Padding: 5px 10px 5px 20px

Margin vs Padding



Display

- Block vs inline-block vs inline



Width vs Height

- Width, Min-width, Max-width
  - Define with px or %
  - When use %, it refer to % of the screen width or the parent element
- Height, Min-height, Max-height
  - Define with px or %
  - When use %, it refer to % of the screen height or the parent element

DIV vs SPAN

- DIV
  - Display: Block
  - Width:100%
  - Like a division/container
  - The contents of div will appear on new line, other than that it will make no difference of the presentation in the page.
- SPAN
  - Display: Inline
  - Like a text wrapper
  - Not support width & height

Exercise

Screen

DIV

Lorem ipsum dolor sit amet, consectetur **adipiscing** elit. Nulla mattis, nisl ut varius euismod, ex nisl tincidunt tortor, pretium mollis diam lorem quis ligula. Praesent a tortor nisl. Nam sed metus lacinia, facilisis massa quis, varius dui.

Screen

DIV

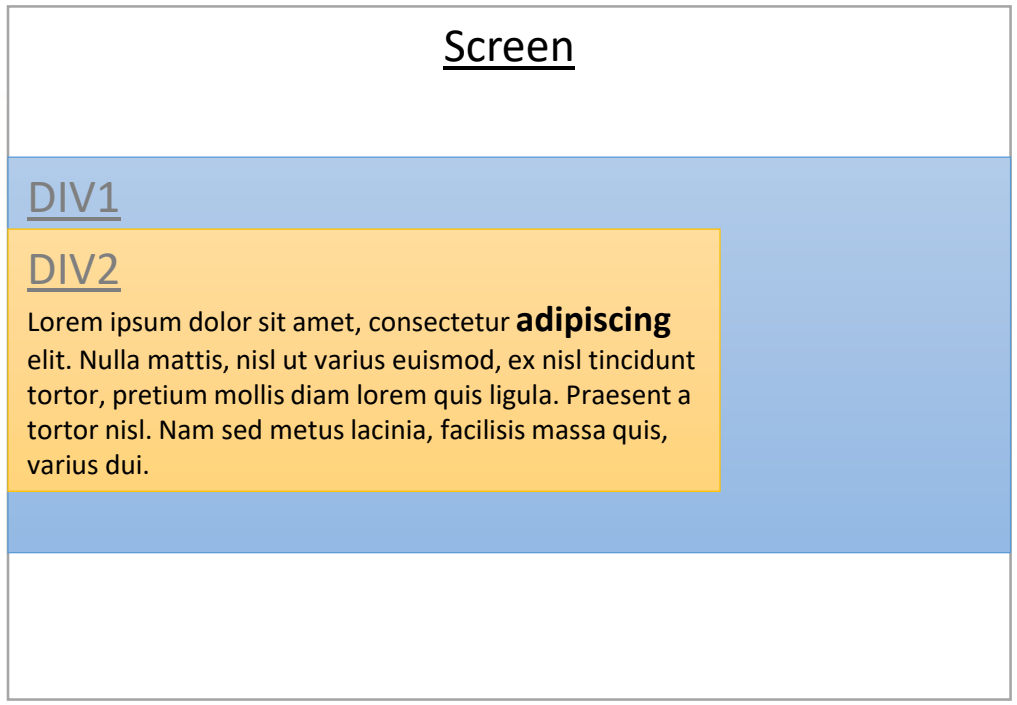
Lorem ipsum dolor sit amet, consectetur **adipiscing** elit. Nulla mattis, nisl ut varius euismod, ex nisl tincidunt tortor, pretium mollis diam lorem quis ligula. Praesent a tortor nisl. Nam sed metus lacinia, facilisis massa quis, varius dui.

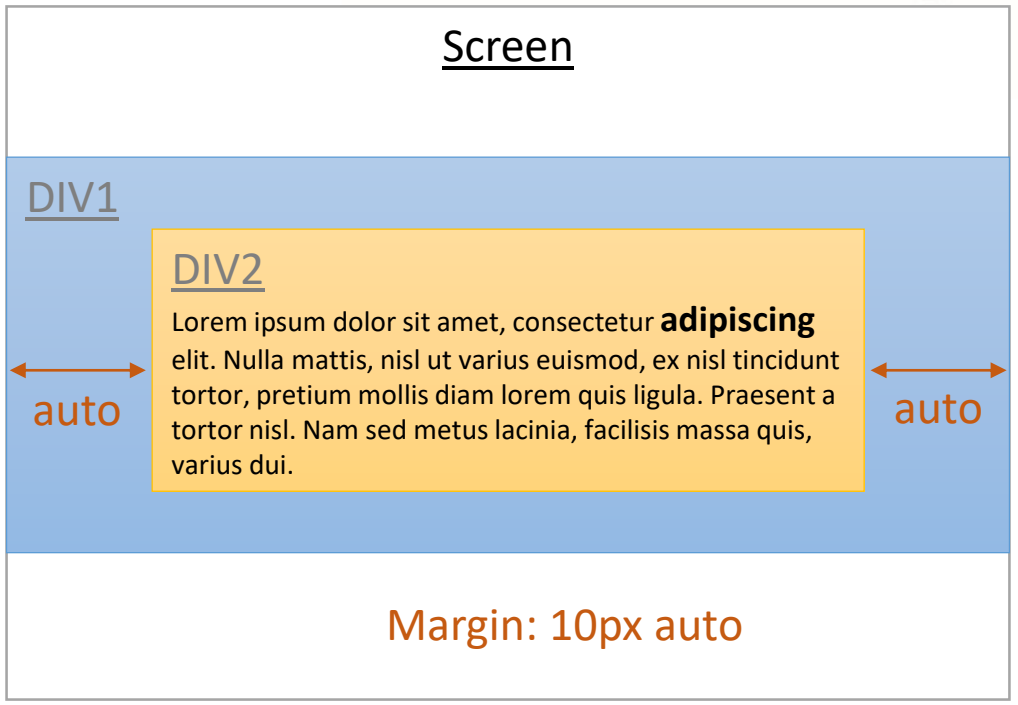
Screen

DIV

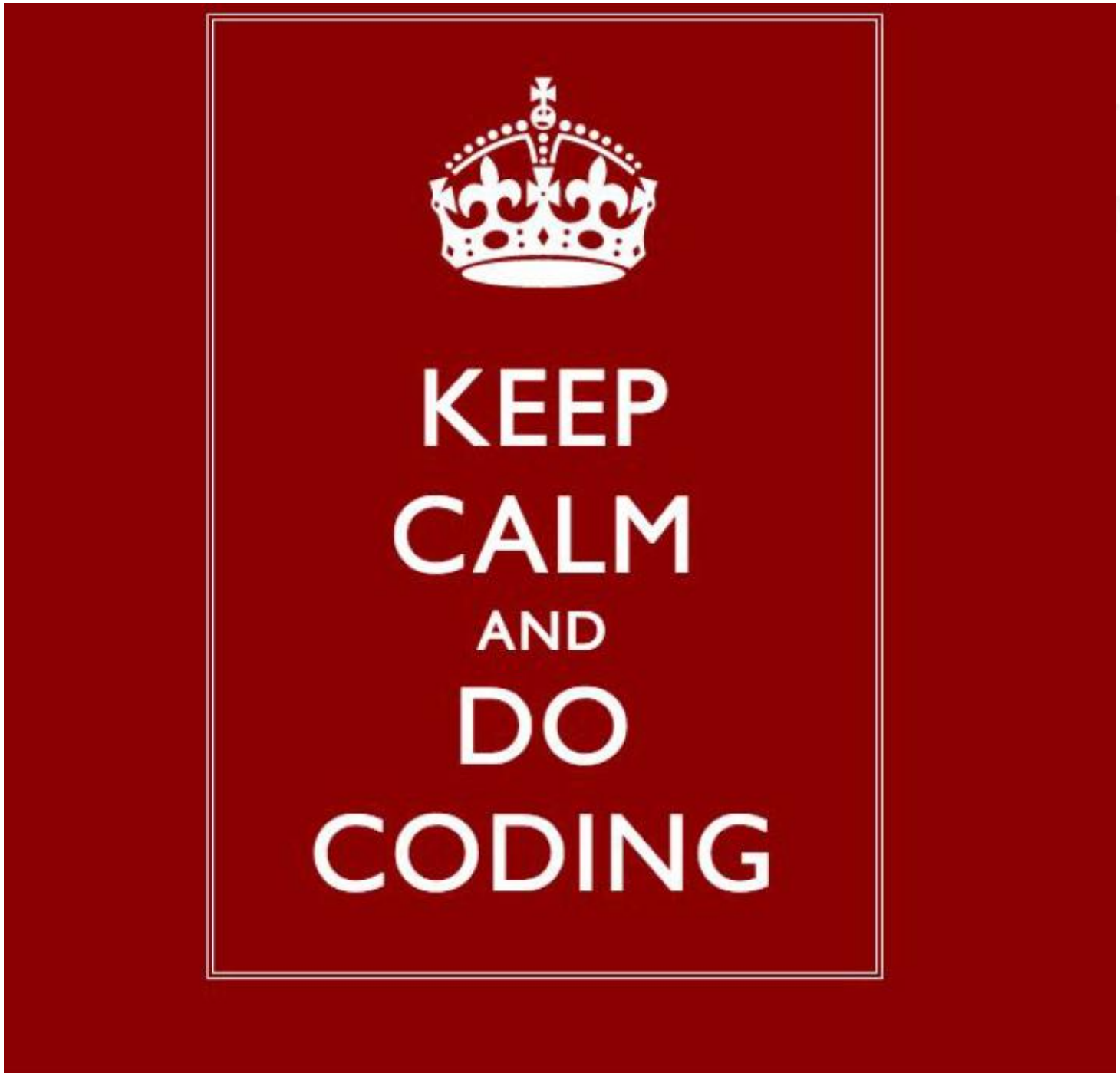
Lorem ipsum dolor sit amet, consectetur **adipiscing** elit. Nulla mattis, nisl ut varius euismod, ex nisl tincidunt tortor, pretium mollis diam lorem quis ligula. Praesent a tortor nisl. Nam sed metus lacinia, facilisis massa quis, varius dui.

Center block





Let’s Code



CSS Selector

- CSS selectors are patterns used to select the elements you want to style.
- Selectors can be accessed by using:
  - **Direct Selectors:** Select element by targeting the ID or class
  - **Relational Selectors :** Target elements based on their relationship to another element within the markup.
  - **Attribute Selectors :** Matching elements based on their attributes.
  - **Pseudo-classes :** match elements based on attributes, user interaction, and form control state
  - **Structural Pseudo-classes :** Target elements based simply on their location in the markup

Direct Selectors

Select by element	Select all <h1> and color it to blue.
	<h1> Hello World</h1>
		H1{
		color: blue;
		}

Select by class	Select all “my-class” class and color it to blue.
	 Hello World
		.my-class{
		color: blue;
		}

Select by ID	Select the “id001” and color it to blue.
	 Hello World
		#id001{
		color: blue;
		}

Relational Selectors

Descendant combinatory (A B)	Select all the “class-B” that are **wrapped within** “class-A”, including child, grandchild, great grandchild.	**.class-A .class-B** {
		color: blue;
		}

Child combinatory (A > B)	Select all the “class-B” that are **direct under** “class-A”,	**.class-A > .class-B** {
		color: blue;
		}

Adjacent sibling, or next sibling selector (A + B)	Select the “class-B” that are **right next to** “class-A”.	**.class-A + .class-B** {
		color: blue;
		}

General sibling or following sibling selector (A ~ B)	Select all the “class-B” that are **same level to** “class-A”	**.class-A ~ .class-B** {
		color: blue;
		}

Exercise

```
<html>
<head></head>
<body>
    <div class="div1">
        <p>Paragraph 1 in the div.</p>
        <p>Paragraph 2 in the div.</p>
        <div class="div2">
            <p>Paragraph 3 in the div.</p>
        </div>
        </div>
    <p>Paragraph 3. Not in a div.</p>
    <p>Paragraph 4. Not in a div.</p>
</body>
</html>
```

- What is the following?
    1. .div1 p
    2. .div1 > p
    3. .div1 + p
    4. .div1 ~ p

Exercise

```
<html>
<head></head>
<body>
    <div class="div1">
        <p>Paragraph 1 in the div.</p>
        <p>Paragraph 2 in the div.</p>
        <div class="div2">
            <p>Paragraph 3 in the div.</p>
        </div>
    </div>
    <p>Paragraph 3. Not in a div.</p>
    <p>Paragraph 4. Not in a div.</p>
</body>
</html>
```

- What is the following?
  1. .div1 p
  2. .div1 > p
  3. .div1 + p
  4. .div1 ~ p

Exercise

```
<html>
<head></head>
<body>
    <div class="div1">
        <p>Paragraph 1 in the div.</p>
        <p>Paragraph 2 in the div.</p>
        <div class="div2">
            <p>Paragraph 3 in the div.</p>
        </div>
    </div>
    <p>Paragraph 3. Not in a div.</p>
    <p>Paragraph 4. Not in a div.</p>
</body>
</html>
```

- What is the following?
  1. .div1 p
  2. .div1 > p
  3. .div1 + p
  4. .div1 ~ p

Exercise

```
<html>
<head></head>
<body>
    <div class="div1">
        <p>Paragraph 1 in the div.</p>
        <p>Paragraph 2 in the div.</p>
        <div class="div2">
            <p>Paragraph 3 in the div.</p>
        </div>
    </div>
    <p>Paragraph 3. Not in a div.</p>
    <p>Paragraph 4. Not in a div.</p>
</body>
</html>
```

- What is the following?
  1. .div1 p
  2. .div1 > p
  3. .div1 + p
  4. .div1 ~ p

Exercise

```
<html>
<head></head>
<body>
    <div class="div1">
        <p>Paragraph 1 in the div.</p>
        <p>Paragraph 2 in the div.</p>
        <div class="div2">
            <p>Paragraph 3 in the div.</p>
        </div>
    </div>
    <p>Paragraph 3. Not in a div.</p>
    <p>Paragraph 4. Not in a div.</p>
</body>
</html>
```

- What is the following?
  1. .div1 p
  2. .div1 > p
  3. .div1 + p
  4. .div1 ~ p

Attribute Selectors

div[attr]	Any element that has the attribute attr regardless of the attribute's value.
div[attr=val]	Element that has a specified attribute and value.
div[attr|=val]	Any element whose attribute attr either has the value val or begins with val- . Commonly used for language.
div[attr~=val]	Any element whose attribute attr has within its value the full word val, surrounded by whitespace.
div[attr^=val]	Any element whose attribute attr starts with the value val.
div[attr$=val]	Matches any element whose attribute attr *ends* in val
div[attr*=val]	Matches any element whose attribute attr matches val any- where within the attribute. It is similar to div[attr~=val], except the val can be part of a word.

Pseudo Classes (1)

button**:enabled**	A user interface element that's enabled, which is basically any form control that supports the disabled attribute but doesn't currently have it applied.
button**:disabled**	A user interface element that is disabled: any form control that supports the disabled attribute and currently has it applied.
button**:checked**	For radio buttons or checkboxes that are selected or ticked.
button**:indeterminate**	For form elements that are neither checked nor unchecked. For example, if you tick a **check all** checkbox to select a group of checkboxes, then deselect some but not all of the checkboxes in the group, the **check all** could be set to the indeterminate state (with JavaScript) to indicate that it's neither checked nor unchecked.
button:**default**	Applies to one or more UI elements that are the default among a set of similar elements. For example, the one radio button in a group of same-named radio buttons that was checked on page load will continue to match :default after another radio button in the same-named group is selected.

Pseudo Classes (2)

Input:**valid**	Applies to elements that are valid, based on the type, pattern, or other input attributes
Input**:invalid**	Applies to empty required elements and elements failing to match the requirements defined by the type or pattern attrib- utes.
Input:**in-range**	Applies to elements with range limitations where the value is within those limitations. This applies, for example, to date/time, number, and range input types with min and max attributes.
Input:**out-of-range**	The opposite of :in-range: elements whose value is *outside* the limitations of their range.
Input:**required**	Applies to form controls that have the required attribute set.
Input:**optional**	Applies to all form controls that *do not* have the required attribute.
Input:**read-only**	Applies to elements whose contents are unable to be altered by the user. This is most elements other than those with the contenteditable attribute set and form fields.
Input:**read-write**	Applies to elements whose contents are user-alterable, such as contenteditable components and writable input fields.

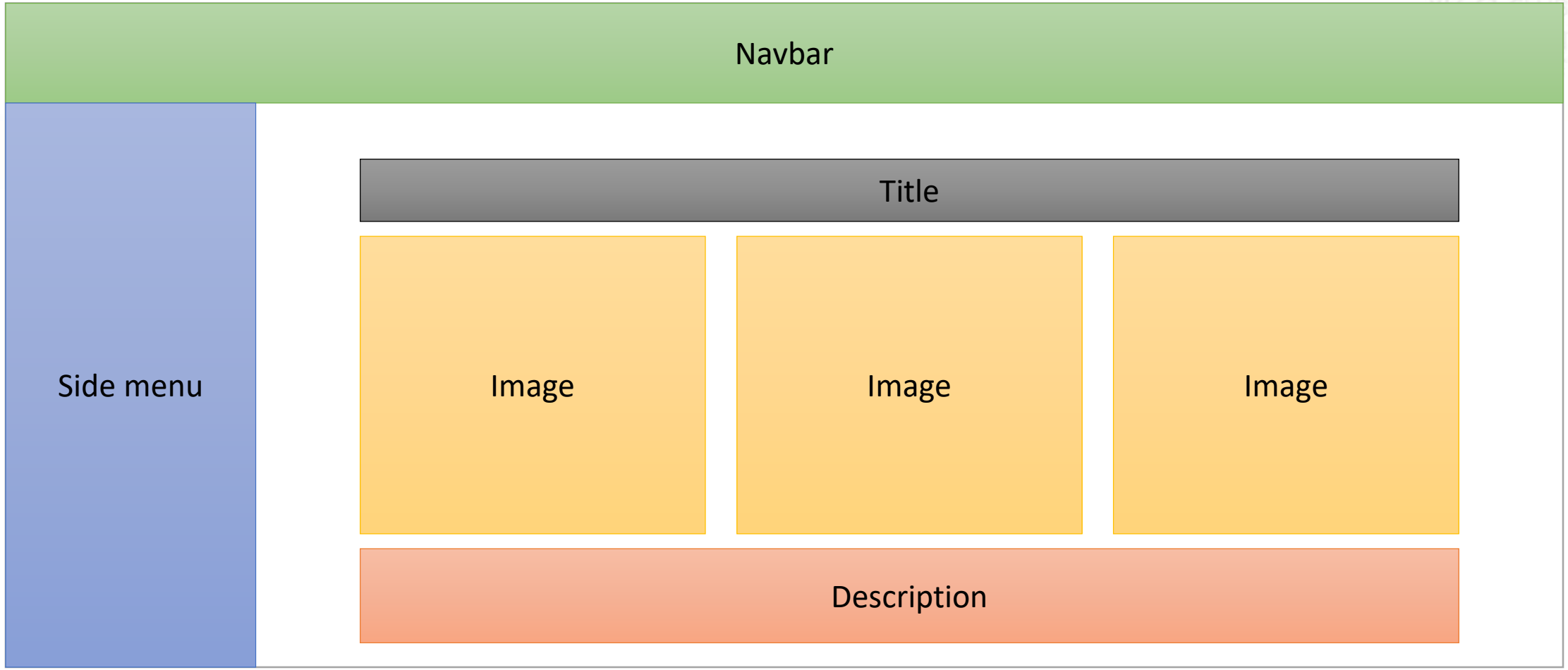
Structural Pseudo Classes (1)

:root	The root element, which is the html element in our HTML files
E:nth-child(n)	The element E that is the nth child of its parent. The n parameter is explained in the note below.
E:nth-last-child(n)	The element F that is the nth child of its parent E, counting backwards from the last one.
E:nth-of-type(n)	The element that is the nth element of its type in a given parent element.
E:nth-last-of-type(n)	Like nth-of-type(n), except counting backwards from the last element in a parent.
E:first-child	The element E if E is the first child of its parent. This is the same as E:nth-child(1).
E:last-child	The element E if E is the last child of its parent, same as E:nth-last-child(1).

Structural Pseudo Classes

E:**first-of-type**	The same as :nth-of-type(1).
E:last-of-type	The same as :nth-last-of-type(1).
E:only-child	Element E if E is the only child of its parent.
E:only-of-type	Element E if E is the only element of type E that is a direct child of its parent element.
E:empty	An element that has no children; this includes text nodes .
E:lang(en)	An element in the language denoted by the two-letter abbreviation, such as en.
E:not(exception)	Select elements that *don't* match the selector in the parentheses.

Structure your page

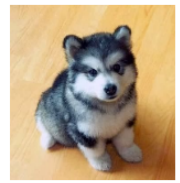


Practice

THE
ANALOG
SPECIALISTS

HOME FOR SALE REPAIRS ABOUT CONTACT

We specialize in the sale and repair of classic keyboards, in particular the Fender Rhodes, Wurlitzer EP200, and Hohner Clavinet.



Jerry Ho
Business Analyst
603.89966799 admin@phd.com.my

Experience

Business Analyst

Qinetics Solutions Sdn Bhd | Kuala Lumpur, Malaysia

Jul 2017 - Present
1 year 2 months

Industry	: Computer / Information Technology (Software)
Specialization	: IT/Computer - Network/System/Database Admin
Role	: Project Management
Position Level	: Junior Executive

Job Description

1. **Requirment Gathering**
   Collect Change Request or New Project requirement before kick start. Diffirent role of people have different way of interpretation of their expectation of new feature or function. Collection it and transform it to technical language so that developer can develop accordingly.

2. **Requirment Analysis**
   Analyse all the requirement and make sure the information is sufficient for developer to develop it. Meanwhile, turn it to a most efficiency workflow or process to save operation cost/development time.

3. **Task Projection**
   Able to report all the current task and project status as well as the Estimated Completion Date.

4. **Task Scheduling**

   Schedule all the task according to the priority and severity as well as business value.

5. **Task Summary & Report**
   Time to time report the project/task status to stakeholders.