Application Systems Development for Business Analytics

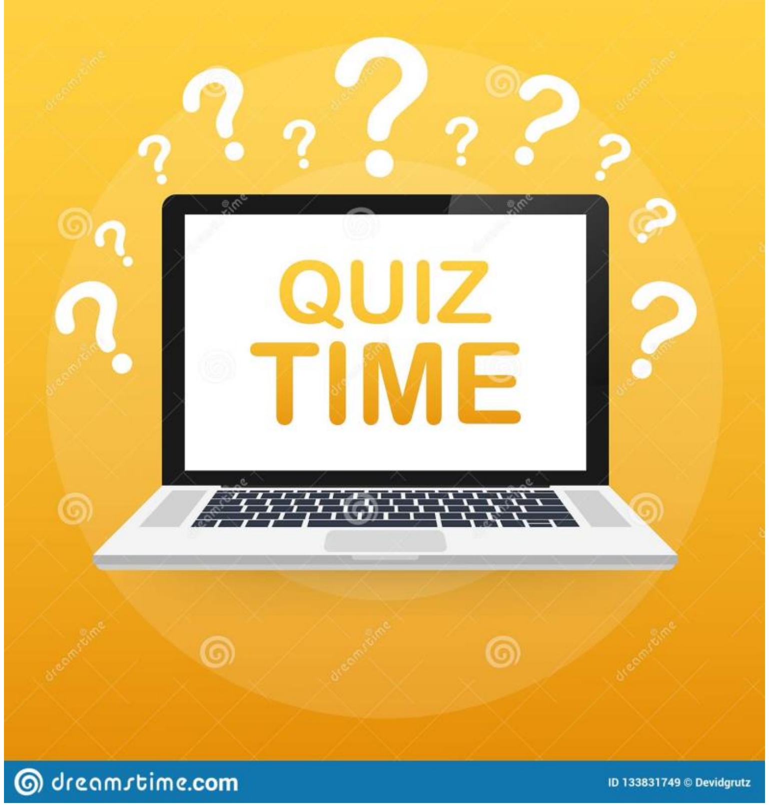
BT3103 - Week 2

2019/2020 Semester 2

Thangamani R

Week 2

- Quiz
- Python Seaborn
- SDLC : Testing , Deployment & Evolution
- HTML & CSS Basics
- Hands on exercises using Discovery tool


QUIZ
TIME

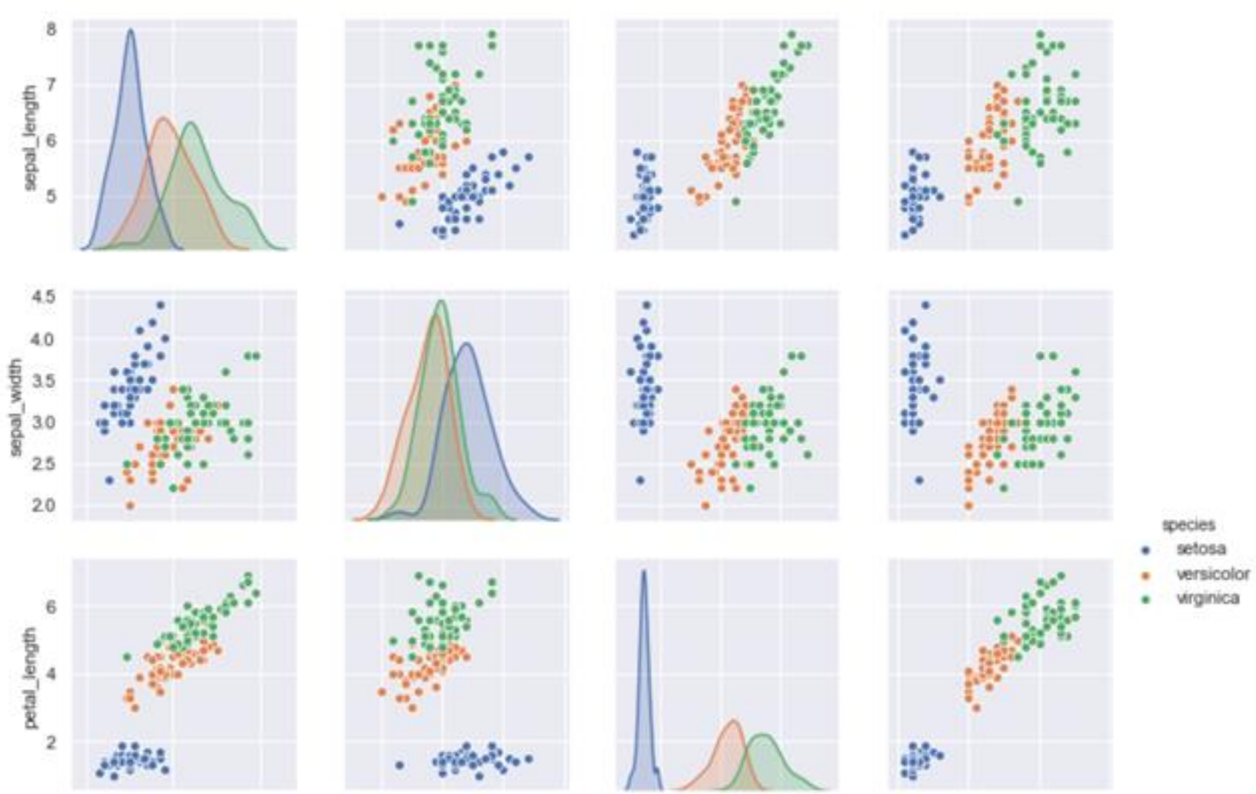
Recap from Week 1

- Jupyter Notebook Installation
  - Installation : https://jupyter.readthedocs.io/en/latest/install.html
  - Recommended using Anaconda
  - Anaconda Reference : https://www.anaconda.com/distribution/
- Text submitted instead of links
  - Text like NA and notes submitted instead of link.

Python Seaborn

- Library for making statistical graphs in Python
- Built on top of matplotlib
- Closely integrated with pandas data structures
- Support for categorical variables
- Convenient views for complex data sets
- Reference :http://seaborn.pydata.org/

Python Seaborn



Python Seaborn

Built in datasets:

- Built in datasets
- *get_dataset_names()* – returns the list of available datasets
- *load_dataset(‘ds_name’)*- loads the specific dataset and returns a dataframe
- https://github.com/mwaskom/seaborn-data

Python Seaborn

In Discovery Tool:

- To load the dataset
- *pd.read_csv(‘Shared_path_for_dataset_name.csv’)*

Seaborn Functions:

- Relational : relplot
- Categorical : catplot

Python Seaborn

Relational:

- Relational plots to depict how variables are related to each other and how the relationship depends on other variables.
- Scatter Plot by default,can be changed using kind attribute.
- *sns.relplot(x=‘var1’,y=‘var2’,data=df,kind=‘plottype’,hue=‘var3’)*

Python Seaborn

Relational:

- Scatterplot - with *kind='scatter'* , default option
- Lineplot can also be drawn - with *kind='line'*
- *Style* and *size* attributes exist

Python Seaborn - Categorical

- Categorical variables(not numerical , divided into discrete groups)
- Eg. Yes/No, Pclass1/Pclass2/Pclass3,Months of the year, Days of the week, Lunch / Dinner etc.
- One of the main variable is categorical , then these plots can be used.

Python Seaborn - Categorical

Syntax:

- *sns.catplot(x=‘var1’,y=‘var2’,data=df,kind=‘plottype’, hue=‘var3’)*
- No style or size attributes

Python Seaborn - Categorical

Categorical scatter plots:

- **`stripplot()`** (with `kind="strip"`; the default)
- **`swarmplot()`** (with `kind="swarm"`)

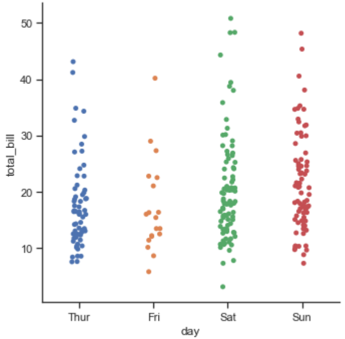
Categorical distribution plots:

- **`boxplot()`** (with `kind="box"`)
- **`violinplot()`** (with `kind="violin"`)
- **`boxenplot()`** (with `kind="boxen"`)

Categorical estimate plots:

- **`pointplot()`** (with `kind="point"`)
- **`barplot()`** (with `kind="bar"`)
- **`countplot()`** (with `kind="count"`)

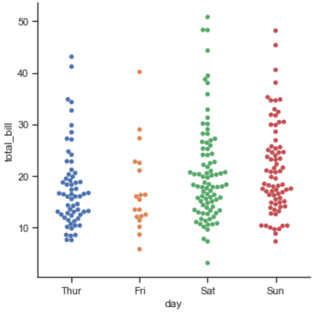
Python Seaborn - Categorical



Strip Plot:

- Strip plot is a scatter plot where one of the variables is categorical .
- Default in catplot.
  - *sns.catplot(x=‘var1’,y=‘var2’,data=df,kind=‘strip’,hue=‘var3’)*

Python Seaborn - Categorical



Swarm Plot:

- Identical to strip plot but prevents the data points from overlapping . Looks like a swarm of bees . Aso known as beeswarm
- Better visualization but works relatively well for smaller datasets.
- *sns.catplot(x=‘var1’,y=‘var2’,data=df,kind=‘swarm’,hue=‘var3’)*

Python Seaborn - Categorical

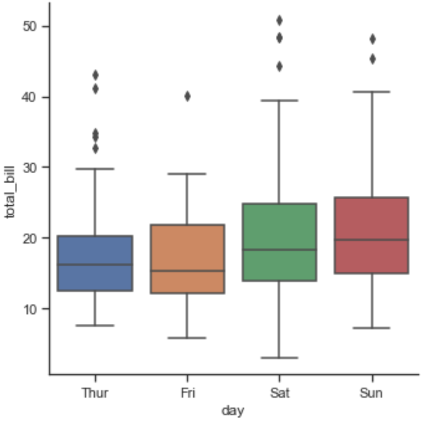
Distribution

- As the size of the dataset grows,, categorical scatter plots become limited in the information they can provide about the distribution of values within each category.
- When this happens, there are several approaches for summarizing the distributional information in ways that facilitate easy comparisons across the category levels.

Python Seaborn - Categorical Distribution

Box Plot

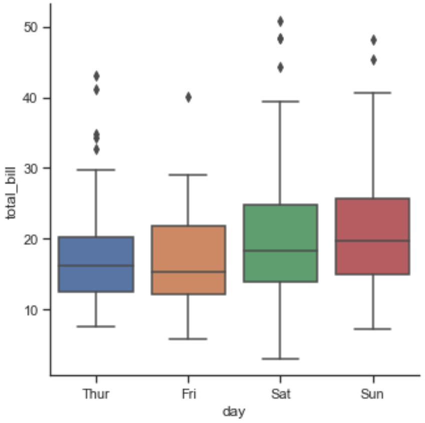
- Shows the three quartile values of the distribution along with the extreme values.
- Center line shows the median value
- Whiskers represent full range of data
- Data outliers can be identified



Python Seaborn - Categorical Distribution

Box Plot

- *sns.catplot(x=‘var1’,y=‘var2’,data=df,kind=‘box’,hue=‘var3’)*



Python Seaborn - Categorical Estimation

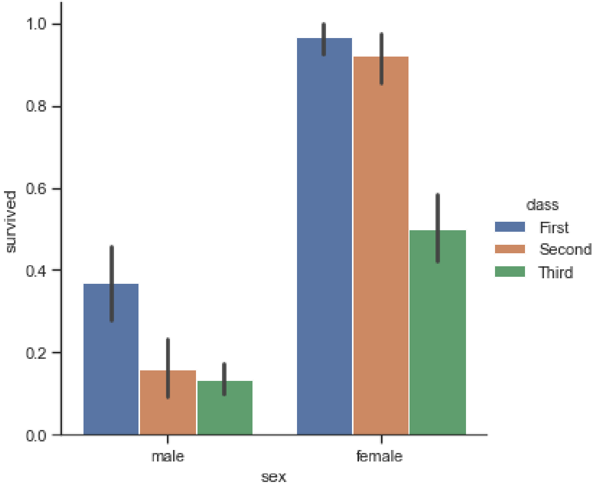
Estimation

- For other applications, rather than showing the distribution within each category, you might want to show an estimate of the central tendency of the values

Python Seaborn - Categorical Estimation

Bar Plot

- Estimation of central distribution
- *sns.catplot(x=‘var1’,y=‘var2’,data=df,kind=‘bar’,hue=‘var3’)*



TRY IT!

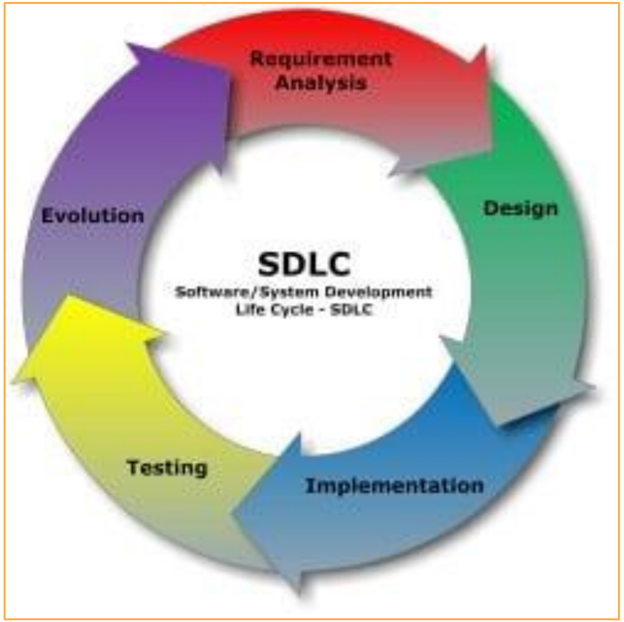
Python Seaborn

- Try out the Seaborn exercises in the Discovery path http://bit.ly/NUS_bt3103

Software Development life cycle

- Write down on what you recollect on SDLC from the previous lecture.

Phases of Software Development life cycle (SDLC)



Phases of SDLC - Development

Unit Testing:

- As part of development, developers need to make sure their code works.
- Unit testing is done on the individual unit of work.
- Unit is smallest testable part

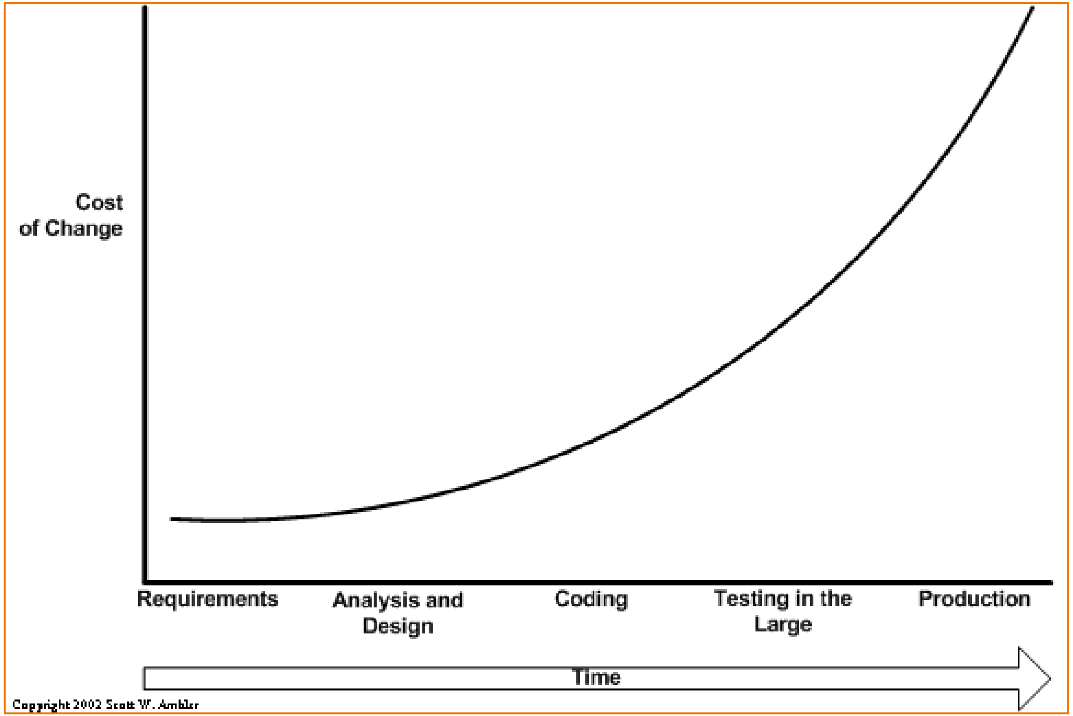
Phases of SDLC - Development

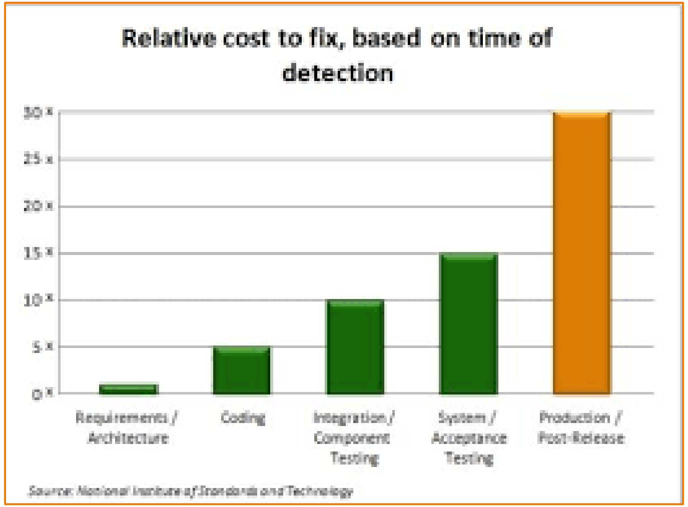
- In procedural programming, a unit may be an individual program, function, procedure, etc.
- In object-oriented programming, the smallest unit is a method.
- Performed prior to Integration testing.

Phases of SDLC - Development

- Several frameworks / tools are used for unit testing . eg. JUnit for java.
- Unit Testing is often neglected but it is one of the most important levels of testing.
- Increases confidence in changing / maintaining code.
- Defects are detected earlier in the SDLC than at a much later phase.

Phases of SDLC - Testing

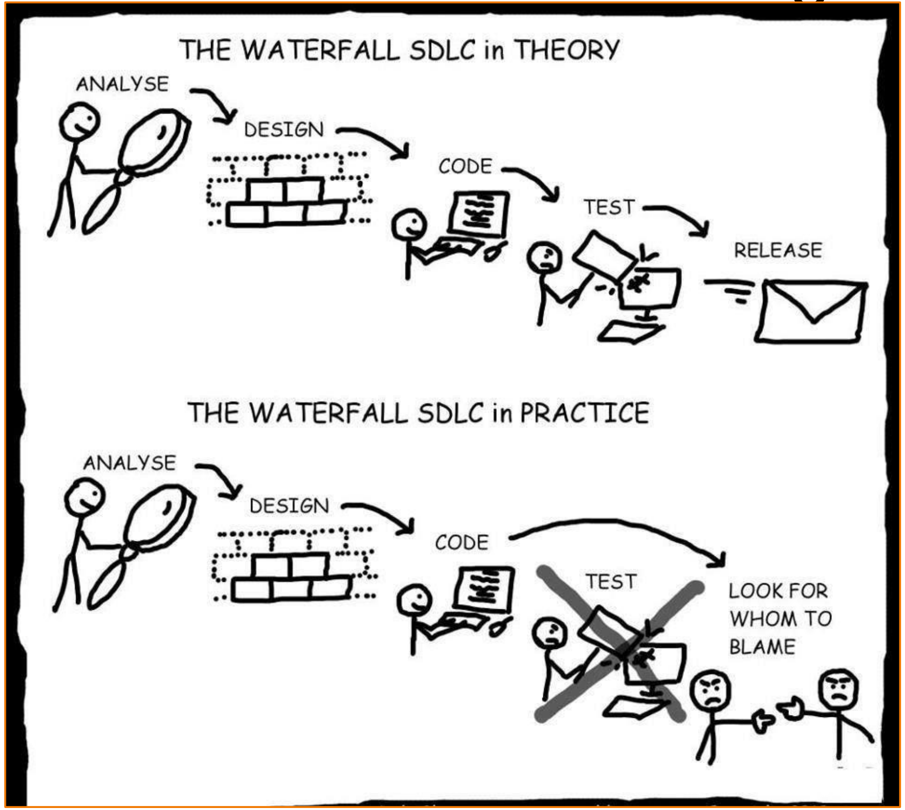


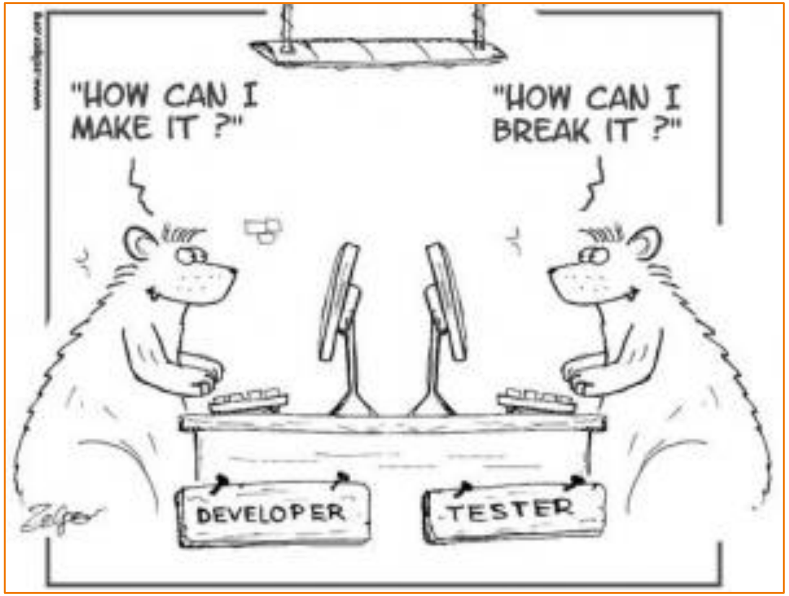


Relative cost to fix, based on time of detection

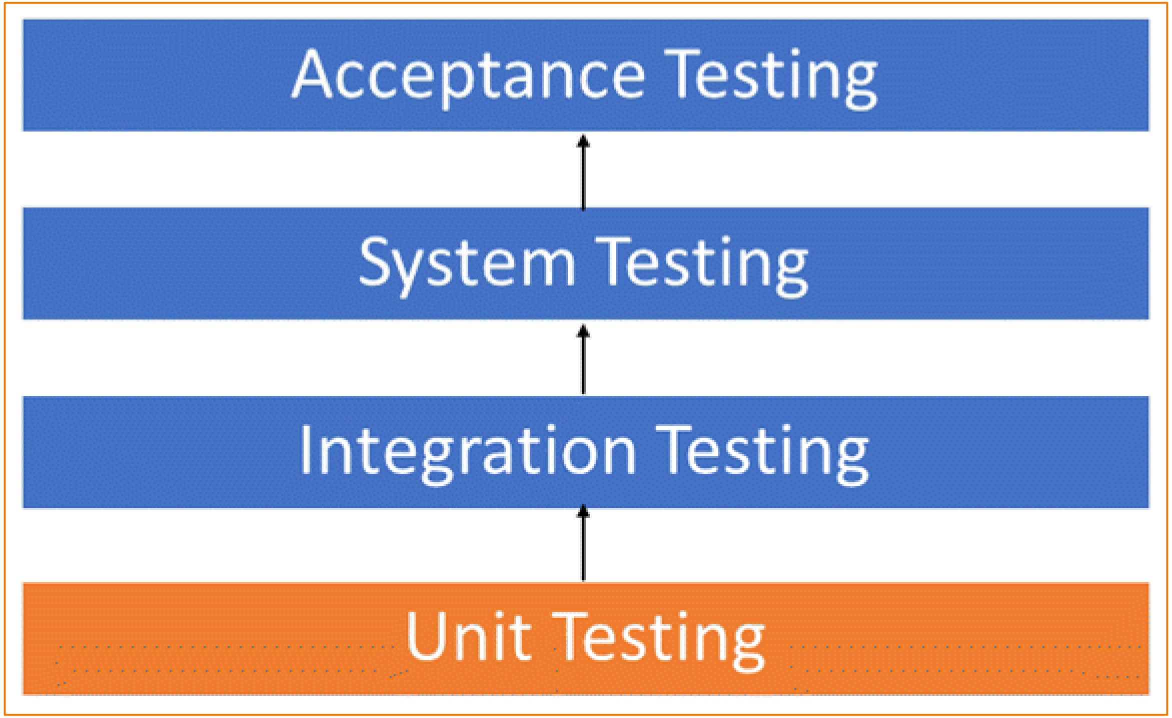
Source: National Institute of Standards and Technology

Phases of SDLC - Testing





Phases of SDLC - Testing



Acceptance Testing

System Testing

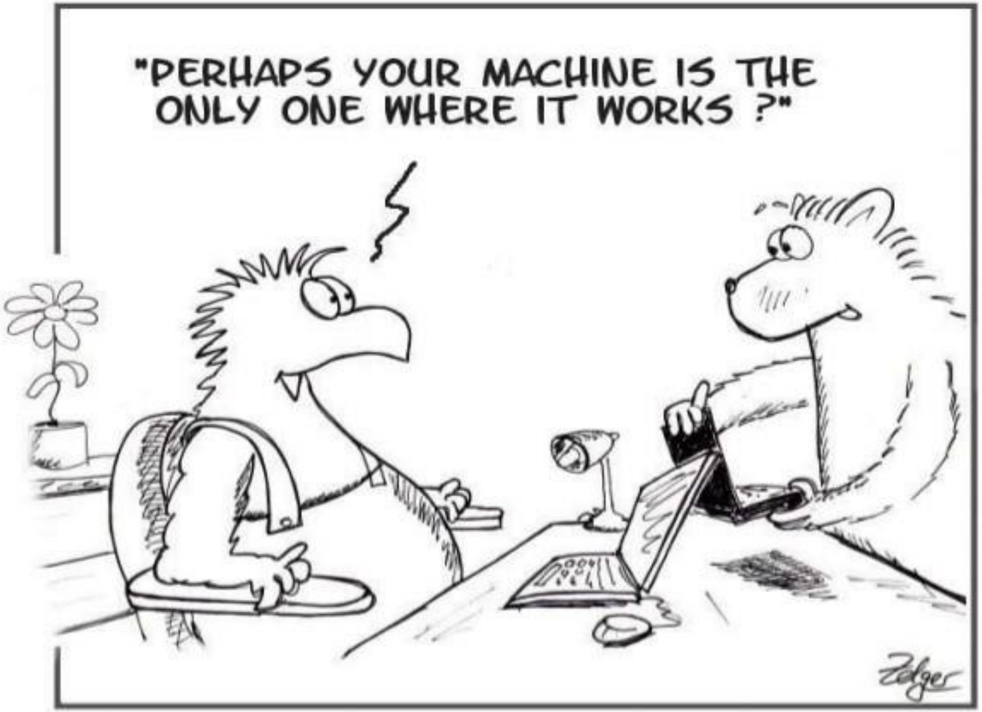
Integration Testing

Unit Testing

Phases of SDLC -Testing

- Software goes through testing phase after development
- Testing environment is different from development environment.
- Different testing approaches are adapted by the organizations to suit their business needs
- Automated Testing or Manual Testing
- Quality Assurance Team ensure that the quality standards are met.

Phases of SDLC - Testing



It works on my machine

Phases of SDLC - Testing

- System is integrated with different components and testing during this phase.
- To make sure system meets the original specifications
- Any defects observed during this phase are fixed and the fixed code/ product is re-tested to make sure no other functionality is impacted by the fix(Regression Test).

Phases of SDLC - Testing

User Acceptance Testing

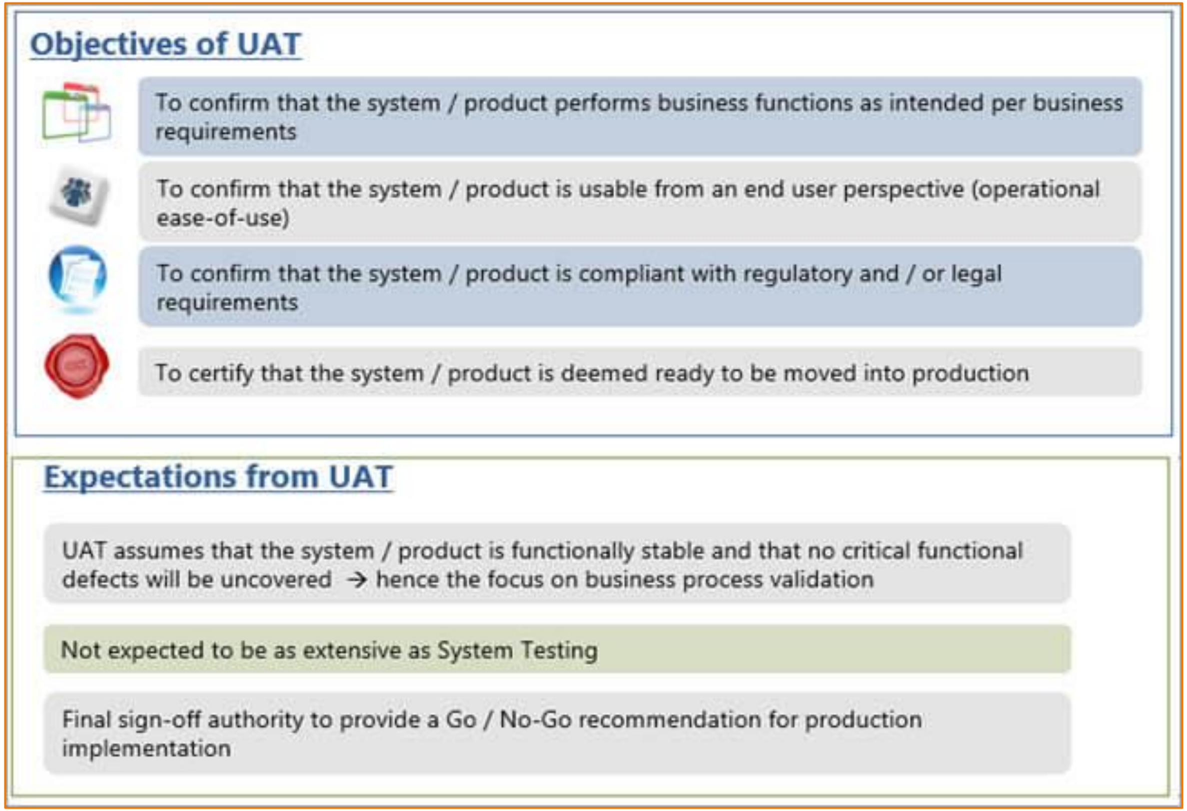
- Actual users of the system are involved in the testing to verify if the developed software is
    - Usable
    - As per requirements
    - Any issues identified

Phases of SDLC - Testing

User Acceptance Testing

- Last phase of testing.
- User sign off is required before the software is released or deployed to production environment.

Phases of SDLC - Testing



Objectives of UAT

- To confirm that the system / product performs business functions as intended per business requirements
- To confirm that the system / product is usable from an end user perspective (operational ease-of-use)
- To confirm that the system / product is compliant with regulatory and / or legal requirements
- To certify that the system / product is deemed ready to be moved into production

Expectations from UAT

- UAT assumes that the system / product is functionally stable and that no critical functional defects will be uncovered → hence the focus on business process validation
- Not expected to be as extensive as System Testing
- Final sign-off authority to provide a Go / No-Go recommendation for production implementation

Phases of SDLC - Deployment

- Software is released in production environment .
- Automated deployment or manual deployment
- Deployment team is involved.
- Usually done after office hours to avoid disruption to the users.
- Software is considered Live.

Phases of SDLC : Evolution

- After deployment software is continuously monitored for performance, functionality , usability etc.
- Based on this improvements are made to the product
- Additional features are developed and subsequently released to production(after going through the SDLC phases)

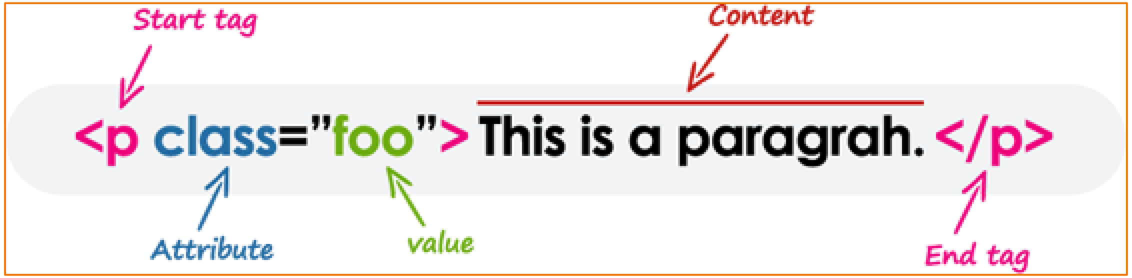
HTML Basics

- Hyper Text Markup Language
- Format on which all the websites are built
- HTML is a format that tells the browser how to display the web page.
- Has the extension .html or .htm
- Predefined tags used.

HTML Basics

- HTML is comprised of elements .
- Elements start and end with a tag.
    - *<tag_name>content</tag_name>*
    - *<tag_name attribute=’value’>content</tag_name>*
- Tags have attributes , values & content
- HTML is not case sensitive(preferential lower case for the tag and attribute names)

HTML Elements



HTML Elements

- DocType Declaration
    - <!DOCTYPE HTML>
    - Must appear at the top of HTML page
    - For the browser to identify the HTML page and render it correctly

HTML Elements

- <html></html>
    - Indicates start of html document
    - Every opening tag has a matching closing tag included </html> in this case.
    - Rest of the page contents should be placed between the opening and closing tags.
    - Only one <html> tag per page

HTML Elements

- <head></head>
  - Header information
  - Contents are not displayed in the browser.
- <body></body>
  - Visible content to be displayed in the browser

HTML Elements

- <title></title>
  - Page title is displayed
  - <title> is included inside the header tag

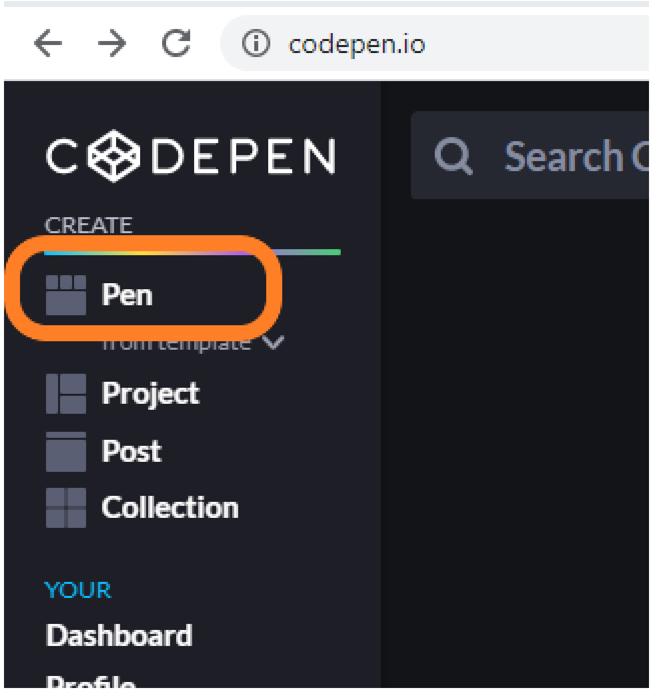
HTML Elements

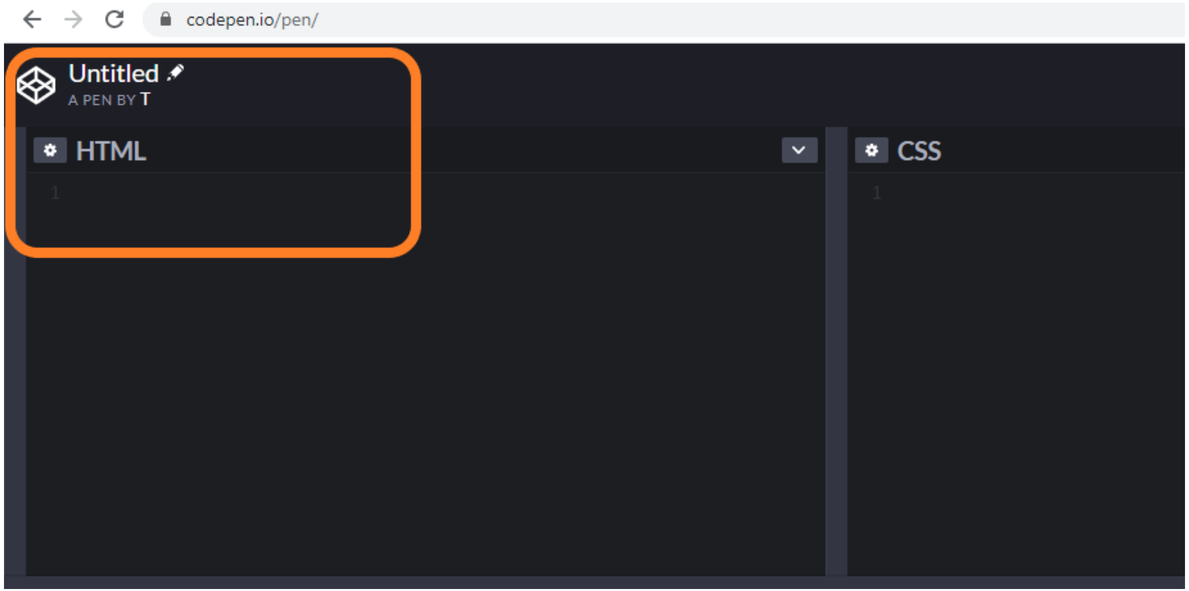
- Headings
  - Has 6 header options from <h1> till <h6>
  - Can be used as headers or subheaders
  - Each heading has a blank line rendered before and after it.

Try it:

- Online Editor : CodePen online editor
  - https://codepen.io/
- Create a html page to display welcome message in bold.

Try it:


codepen.io
CODEPEN
Search C
CREATE
Pen
from template
Project
Post
Collection
YOUR
Dashboard
Profile


codepen.io/pen/
Untitled
A PEN BY T
HTML
CSS

HTML Comments

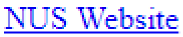
- <!-- This is a html comment -->
- Used to add notes, explanations inside a document.
- Comments are ignored by the browser.

HTML Tags

- <div></div>
- Division or section in a document
- Used as a container for other elements

HTML: Anchor Tag

- To create link to another document or web page.
- Syntax
  - `<a href=”url”>Text to be displayed</a>`
- Eg.
  - `<a href=”http://www.nus.edu.sg/”>NUS Website</a>`



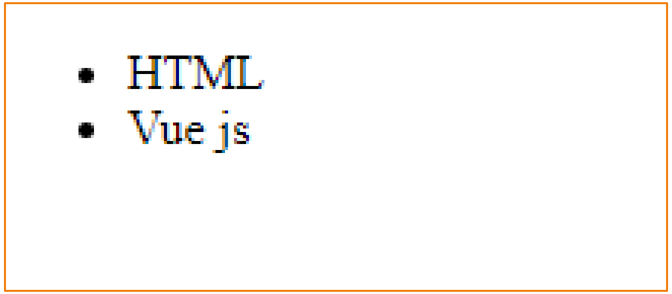
NUS Website

HTML: Image Tag

- To display an Image.
- Syntax
  - *<img src="Image path" alt="Text">*
  - *Alt text will be displayed if the image cannot be found.*
- Eg. *<img src="https://i.gifer.com/TDwQ.gif" alt="Chef Image">*

HTML: Lists

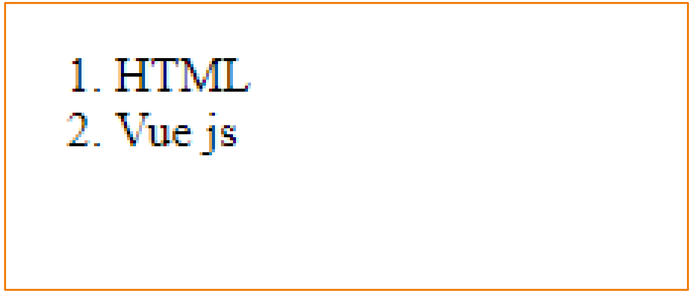
- Unordered Lists
- Syntax
  - *`<ul><li>item 1</li><li>item 2</li></ul>.`*
- Eg.
  - *`<ul><li>HTML</li><li>Vue js</li></ul>`*



- HTML
- Vue js

HTML: Lists

- Ordered Lists
- Syntax
  - `<ol><li>item 1</li><li>item 2</li></ol>.`
- Eg.
  - `<ol><li>HTML</li><li>Vue js</li></ol>`



1. HTML
2. Vue js

Try it:

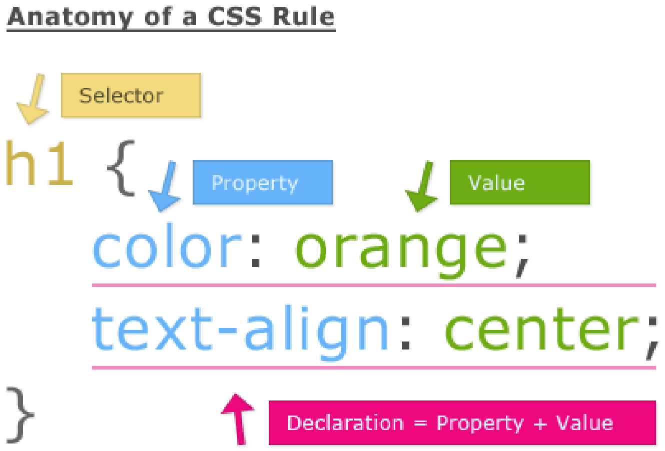
- Online Editor : CodePen online editor
  - https://codepen.io/
- Create a html page
  - With link to another website
  - With an Image
  - Ordered and Unordered list of items

Cascading Style Sheets (CSS)

- To separate page content from appearance
- Applying style rules to change appearance of the webpage.
- Comprises of Selector and Style rule.
- https://developer.mozilla.org/en-US/docs/Web/CSS/Reference

Cascading Style Sheets (CSS)

- For eg. , applying a style to h1 element.



Anatomy of a CSS Rule

```
h1 {
    color: orange;
    text-align: center;
}
```

Cascading Style Sheets (CSS)

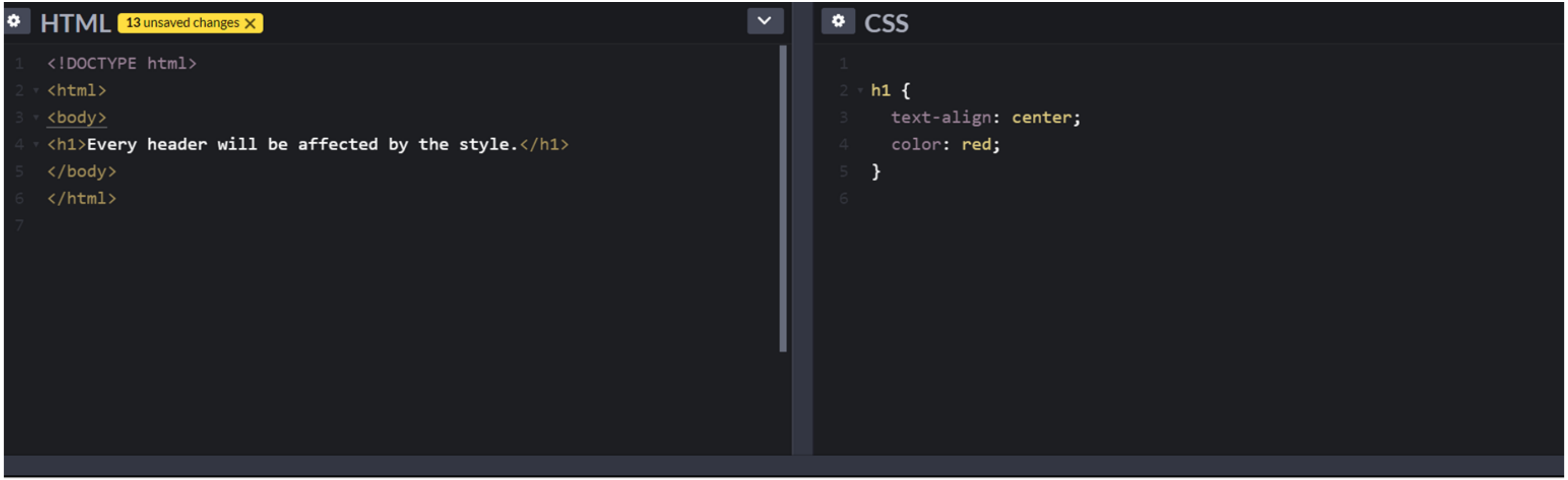
- External Style Sheets
  - CSS files are text files with .css extensions
  - With external style sheets the HTML code is separated from the styling details and hence the code is easier to maintain.
  - CSS file is then linked to the HTML file.

Cascading Style Sheets (CSS)

- Linking Style Sheets
  - *<link type=”text/css” rel=”stylesheet” href=”css_path”/>*

democss.css

CSS - By Element



```html
<!DOCTYPE html>
<html>
<body>
<h1>Every header will be affected by the style.</h1>
</body>
</html>
```

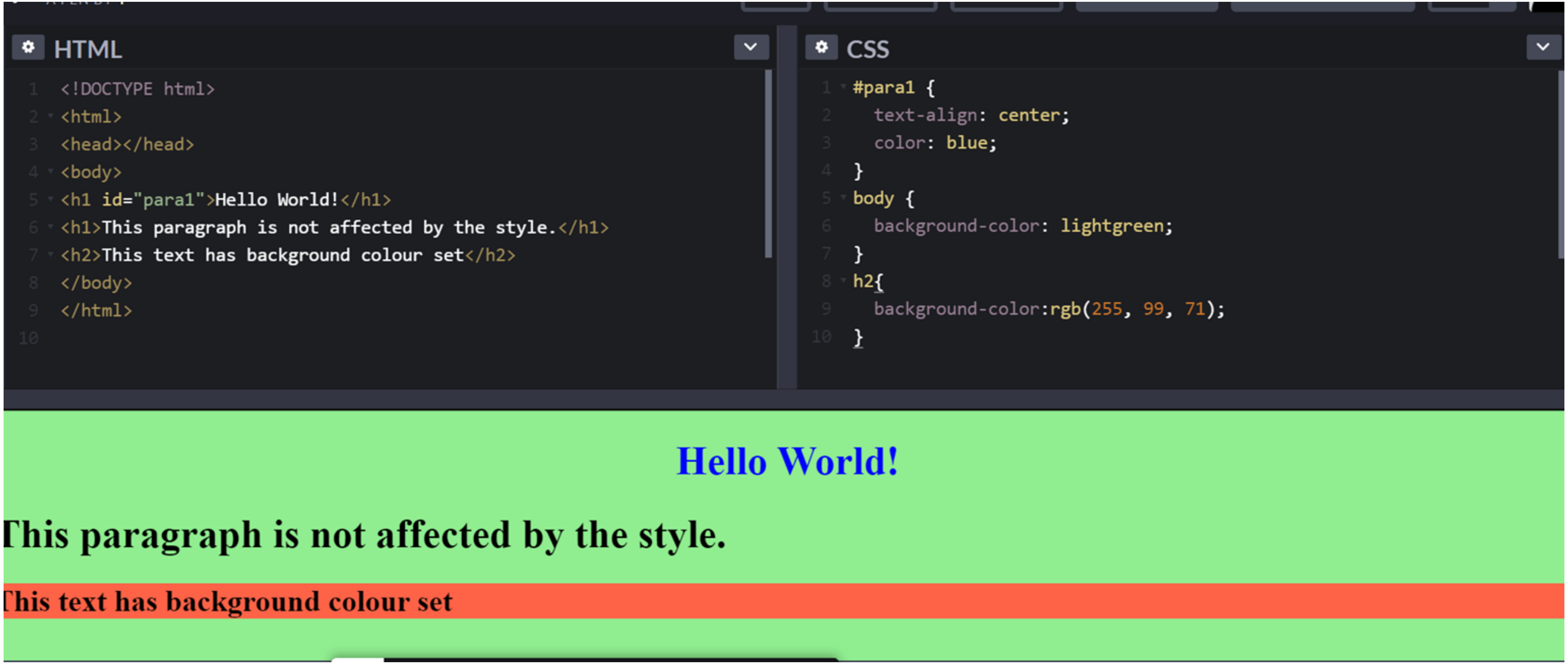
```css

h1 {
  text-align: center;
  color: red;
}
```

Every header will be affected by the style.

CSS - Example

democss.css

demopage.html



```html
<!DOCTYPE html>
<html>
<head></head>
<body>
<h1 id="para1">Hello World!</h1>
<h1>This paragraph is not affected by the style.</h1>
<h2>This text has background colour set</h2>
</body>
</html>
```

```css
#para1 {
  text-align: center;
  color: blue;
}
body {
  background-color: lightgreen;
}
h2{
  background-color:rgb(255, 99, 71);
}
```

Hello World!

This paragraph is not affected by the style.

This text has background colour set

Try it:

- Online Editor : CodePen online editor
  - https://codepen.io/
- Appy CSS style sheet for the previously created HTML file
  - Add background colour
  - Add text highlighting
  - Add colours for your lists

Wrap up:

What was covered:

a. Seaborn
b. SDLC (Testing , Deployment & Evolution)
c. HTML & CSS Basics

Homework Problems

1. **Requirements:**
    a. Last week as part of the requirements gathering you had listed down a few applications which you think would be helpful to students. Pick one application from the list and design an User Interface using paper and pen. Take a picture of the UI, upload to google drive and share the link in Discovery tool.

Homework Problems

2. **Create HTML file in your local machine:**
    a. Use any text editor of your choice and create a html file in your local machine.
    b. How do you get to view your the html page(result of your code) in your machine?
    c. Try out the various elements discussed in the class.