

# SD Class

## Course Topics

- Python
  - Basic Features
  - Anatomy of a python program
  - Modules, Packages
  - Zen of Python
  - How to learn a new language by yourself?
  - Exercise
- Command Line, Terminal, Shell, Bash
  - Why use command line?
  - Common commands
  - Manual Pages
  - Installing new commands
  - Configurability
  - Command line tip of the day, every day
- Automated Testing
  - What is automated testing, why use it?
  - How to use it?
  - Test Driven Development
  - Testing Setup for Python
- Version Control using git
  - What is version control, why use it
  - How does git work
  - Basic git commands
- How does a website/app work?
  - What all happens when you a search word in Google and press Enter?
  - What is server?
  - HTTP, HTTPS
  - Web frameworks
- Django
  - Model View Controller, urls.py
  - ORM and database
  - Anatomy of a Django Project
  - settings.py
  - requirements.txt
  - Your first view
  - Running locally
- Django Project
  - Templating
  - CRUD Views
  - Authentication
  - Production setup

- Logging and Email
- Designing UI
  - Javascript, CSS, HTML
  - Frontend frameworks
- Using HTTP APIs
- Hosting a website online
  - Cloud Service Providers
  - Load Balancer, nginx
  - Multi Server setup
  - Deploy tools
  - Monitoring
  - Continuous Integration
- Security, Privacy
- Designing Responsible Software

## Session 1

### Introductions

### Python Features

- How to run a program
  - python filename.py
  - Python interpreter. When is it useful?
  - Compare it to other languages
- python hello.py

```
print("Hello Stemland")
```

- Declaring Variables. Dynamically Typed language vs Statically typed language

```
1 # Integer
2 some_number = 12
3
4 # String
5 planet_a = "earth" # Can also use single quote
6
7 # Lists
8 planets = ["pluto", "saturn", "venus"]
9 print(planets)
10 print(planets[1])
11 print(planets[1:2])
```

```

12 print(planets[10])
13
14 # Dictionary
15 capital_by_state = {
16     "Tamil Nadu": "Chennai",
17     "Telangana": "Hyderabad"
18 }
19 print(capital_by_state["Telangana"])
20
21 # Boolean
22 is_it_sunday = False
23

```

- Operators

```

1 print(2 * 5)
2 print(2 ** 5)
3 print(100 % 3)
4 print("Hello " + "World")

```

- Statements

```

1 # If Else statements
2 a = 5
3 if a % 2 == 0:
4     print("Even number")
5 else:
6     print("Odd number")
7
8 # Functions
9 def max(a, b):
10     if a < b:
11         return b
12     elif a > b:
13         return a
14     else:
15         raise Exception("Both are equal")
16
17 # for statement
18 for x in [1, 3, 5]:

```

```
19 print x
20
21 # while statement
22 num = 100
23 while num < 1:
24     num = num / 2
```

### Exercise 1

- Write a function that takes a list of numbers and returns the biggest number.
- Inbuilt functions/keywords

```
1 print("Hello")
2 raw_input("Give me some input")
3 import, from, as
4 dir
```

## Modules

- Namespace means what keywords are available to use in a particular context
- Global vs private variables
- Any function/variable/class defined in one file can be imported into any other python file in the same directory

```
1 from math_utils import *
2 # or
3 from math_utils import max_number
4 max_number(10, 17)
5 # or
6 import math_utils
7 math_utils.max_number(10, 17)
```

- Inbuilt modules
  - datetime, sys
  - file IO
  - urllib, etc
- Figuring out usage of a module
  - Find out the docs for that module
  - It shows usage and examples
  - <https://docs.python.org/2/library/datetime.html>
- Passing arguments

```
1 import sys
2
3 def say_hello(name):
4     print(name)
```

```
5  
6 say_hello(sys.argv[1])
```

## Testing

- What is automated testing, why use it?
- How to use it?
- Test Driven Development

## Package Management

- pip is the packaging software for Python
- Usual standard is that all packages for a software are defined inside a requirements.txt file
- virtualenv can be used to create an environment specific to a project

### Exercise 2

- **Guess the random number:** Write an interactive command line game that picks a random number and asks the user to guess, giving hints each time if number is smaller or bigger

## Zen of Python

Philosophy or style of the language is based on set of principles. So when we write new code we should keep that in mind.

[https://en.wikipedia.org/wiki/Zen\\_of\\_Python](https://en.wikipedia.org/wiki/Zen_of_Python)

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.

Tools to achieve that

- Follow a styleguide and be consistent. e.g. Google's Python Styleguide  
<https://google.github.io/styleguide/pyguide.html>
- Install a Lint module on your IDE that highlights when you make a mistake.
  - <http://flake8.pycqa.org/en/latest/> or <http://pep8.readthedocs.io/en/release-1.7.x/intro.html>
- Code reviews

## Anatomy of a python program

- shebang
- imports
- constant declarations
- Classes, functions
- statements

## How to learn a new language?

- First, get a good understanding of underlying programming concepts using any one low level language like C via a full online course like [CS50x](#)
- Usually three resources
  - Language specific online courses: e.g. [MIT Course on Python](#)
  - Tutorial websites that explains using examples: <https://www.learnpython.org/>
  - Reference documentation for the language: <https://docs.python.org/3/>

## Command line

- Why use command line?
  - Server and website administration
  - Ability to combine them via pipes
  - Faster
- Common commands
  - pwd, cd, mkdir, cp, mv
  - ssh, scp
  - cat, grep, less
  - python, virtualenv, pip
- How to learn
  - Man pages
  - Always use it for development
  -
- Installing new commands using package manager
- Configurability
- Command line tip of the day, every day

### Command line tips - 1

#### Keyboard shortcuts

- Ctrl-a - go to beginning of a line
- Ctrl-e - go to end
- Ctrl-Navigation-keys - Jump words
- Ctrl-R - search and autocomplete previously typed commands

## Take Home Exercise

Hangman <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/assignments/>

- Follow the instructions in the assignment
- Write as many functions as you can, but always write tests before