

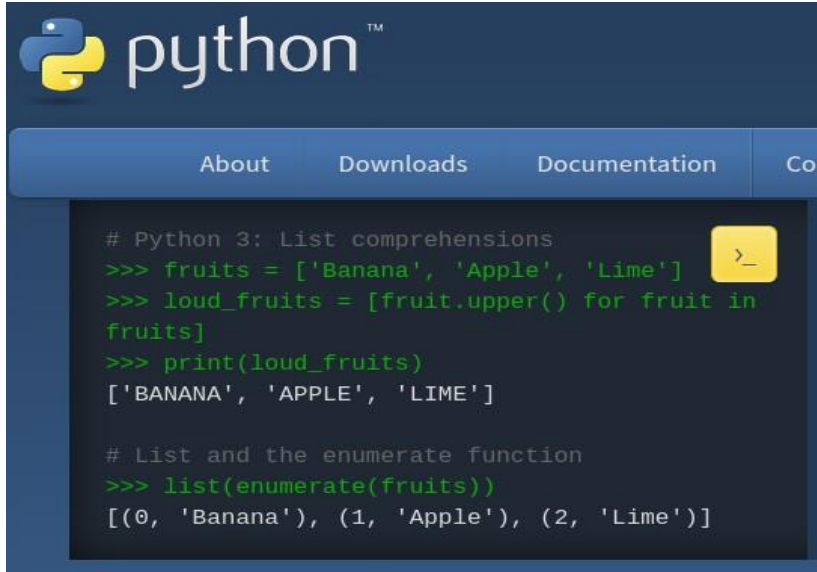
# Discrete Structures!

CMPSC 102



ALLEGHENY COLLEGE

# Get Python3



- Get Python3 from the Python Software Foundation:

<https://www.python.org/downloads/>

# Install Your Own Python3



- Download and install the version of Python3 for your OS being sure to add the PATH to the environmental variables (check the path option!)
- Check with the installation material to learn how to launch Python3

# Running the Python3 Shell

- Type *python3* in the terminal to start the Python shell.
- Type statements or expressions at prompt:
  - `print("Hello, world")`
  - `x = 12**2`
  - `print(x)`
  - `print(x/2)`
  - `# bla bla bla...`
  - `# (This is a comment: everything after the # is ignored )`
- To exit the Python shell, type: `exit()` or Ctrl + Z to exit.

# Data Types

- Integers, counting numbers
  - `num_int = 1`
- Floats, decimals
  - `num_float = 3.1415`
- Strings:
  - `s_str = "Hello World"`

# Data Types

```
height_int = 5
print(f" The height is: {height_int}")
print(" The height is:", height_int) # print another way
```

```
num_float = 3.14
print(f" The float variable is : {num_float}")
```

```
s_str = "Hello World"
```

```
print(" The integer is equal to: ", s_str)
```

# Key Components

All programs built out of

- **Function calls:** Granting temporary kernel-time and/or using issuing parameters to a sub-sequence of instruction in a program.
- **Assignment statements:** The issuing of a value to a variable or place in memory to contain the value.
- **Iteration constructs:** Structures used in computer programming to repeat the same computer code multiple times (*loops*).
- **Conditional logic:** the use of logical rules in code to govern steps taken.
- **Variable creation:** The introduction of an object in memory to contain some value.
- **Variable computations:** The use of values contained in variables to create new value using an operator.
- **Variable output:** The revealing of some value in a variable by printing or another means.

# Key Components

- **Function calls:** Granting temporary kernel-time and/or using issuing parameters to a sub-sequence of instruction in a program.

```
def greet(name):  
    print(f"Hello, {name}! Welcome to Python.")  
  
greet("Alice")
```

- **Assignment statements:** The issuing of a value to a variable or place in memory to contain the value.

```
x = 10  
name = "Alice"
```



# Key Components

- **Iteration constructs:** Structures used in computer programming to repeat the same computer code multiple times (*loops*).

```
fruits = ["apple", "banana", "cherry"]
```

```
for fruit in fruits:  
    print(fruit)
```

- **Conditional logic:** the use of logical rules in code to govern steps taken.

```
age = 18
```

```
if age >= 18:  
    print("You are an adult.")
```

# Key Components

- **Variable computations:** The use of values contained in variables to create new value using an operator.

```
a = 10
```

```
b = 5
```

```
sum_result = a + b    # Addition
```

```
diff_result = a - b    # Subtraction
```

```
prod_result = a * b    # Multiplication
```

```
quotient_result = a / b # Division (float)
```

```
mod_result = a % b     # Modulus (remainder)
```

```
exp_result = a ** b    # Exponentiation (10^5)
```

## Application - Using Python to Find a Name in a File

```
file = open("name")
for line in file:
    if line.startswith("John"):
        print(line)
```

- Can you explain the behavior of this program segment?
- What are the **constructs** inside of this program segment?
- What is the purpose of the *open* function?
- What is the purpose of the *line.startswith* function?

# Application: Using Python to Find an Email in a File

```
file = open("emails")
for line in file:
    name, email = line.split(",")
    if name == "John Davis":
        print(email)
```

- Can you explain the behavior of this program segment?
- What are the **constructs** inside of this program segment?
- What is the purpose of the *open* function?
- What is the purpose of the *line.split* function?

# Runnable Application: Using Python to Find an Email in a File

```
#!/usr/bin/env python3
# """ Demo program """
```

```
myFile_list=["Bob Bye,bob@big.com", "Julie Roth,Jroth@thinktank.com", "John Davis,
JDavis@KingOfTheWorld.com"]
print(f"\n Opening myFile :{myFile_str}") # file = open("emails")
for line in myFile_list:
    print(f"\t + line : {line}, {type(line)}")
    name, email = line.split(",")
    if name == "John Davis":
        print(f"\tName found: {email}")
```

- Can you explain the behavior of this program segment?
- What are the **constructs** inside of this program segment?

# Runnable Application: Using Python to Find an Email in a File

```
#!/usr/bin/env python3 """ Demo program """  
mylist =[  
    "Bob Bye,bob@big.com",  
    "Julie Roth,Jroth@thinktank.com",  
    "John Davis,JDavis@KingOfTheWorld.com",  
    "Tylor Swift,tSwift@Swifter.com",  
    "The Hulk,greenThumb@gardeningHelp.com",  
    "Sherlock Holmes,sHolmes@consultingDetective.com"  
]  
print( "\n Opening mylist :{mylist}")  
for line in mylist:  
    print(f"\t + line : {line}, {type(line)}")  
    name, email = line.split(",")  
    if name == "John Davis":  
        print(f"\t Name found: {email}")  
    if "Sherlock" in name:  
        print(f"\t Detective's Name found: {email}")
```

# Runnable Application: Using Python to Process and Analyze Data

```
#!/usr/bin/env python3  
""" Demo program """
```

```
myFile = [1,2,3,4,5,6,7,8,9,10]  
sum = 0  
count = 0  
for line in myFile:  
    n = int(line)  
    sum += n  
    count += 1  
print(sum/count)
```

- Can you explain the behavior of this program segment?
- What are the **constructs** inside of this program segment?

# Runnable Application: Using Python to Process and Analyze Data

```
#!/usr/bin/env python3
""" Demo program """

sum = 0
count = 0
myFile = open("data.txt")
for line in myFile:
    n = int(line)
    sum += n
    count += 1
print(sum/count)
```

- What are the contents of the data.txt file?
- What is the purpose of the for line in file statement?