



Lecture 3

Introduction to Functions

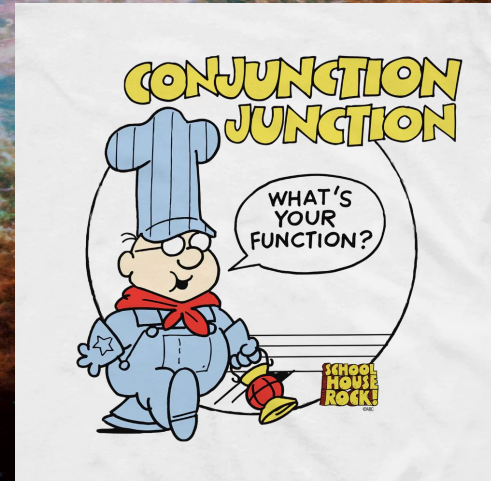
The background of the slide is a deep space image showing the cosmic web, a vast network of filaments and voids of matter and energy. The filaments are colored in vibrant shades of orange, red, and yellow, while the voids are a deep, dark blue. The overall effect is a complex, web-like structure that fills the frame.

Workshop 1 Review

What is a function?

- Much like any number of mathematical functions you know about
- Takes in an independent variable(s) and returns the result of the dependent variable(s)
- Built-in functions
 - `print`, `abs`,
`max`,
`min`

```
def func(x):  
    y = x**100 + 23*x**2 + 56  
    return y
```



How to write a function?

```
def calc_dist(dx,dy):  
    dist = sqrt(dx**2 + dy**2)  
    return dist
```

Define statement

Function Name

Arguments

Function Body

Return Statement

How to Use a Function

- The name of your function is what you will use to call it later in your code
- You will also need to define the variables for your function to use

```
def magic_guessing_game(k):  
    double = 2 * n  
    plus_k = double + k  
    divide_by_2 = plus_k / 2  
    subtract_original_number = divide_by_2 - n  
    print("The number you have now is:" + str(subtract_original_number))  
    return subtract_original_number  
  
# Pick a number 1-10, in this case we pick 7  
n = 7  
abracadabra = magic_guessing_game(12)  
abracadabra
```


Defining Arguments

- Your argument is one of the most important parts of your function
- It defines the function's input, content, and output
- If you know what value will be used most of the time for an argument, you can cheat and pre-define it
- Watch your argument type, this will be where a lot of your errors come from

```
In [8]: 1 def distance( x, y=0. ):
        2     return abs(x-y)
        3
```


Functions True/False

- For each of the following snippets of code, answer True if you believe the code will work, or answer False if you believe the code will error for any reason

```
def add_one:(n)  
    return n + 1
```

```
add_one(7)
```

```
def miles_to_feet(miles):  
    return miles * 5280
```

```
miles_to_feet(26.2)
```

```
def area_of_circle_with_radius(r):  
    r = area_of_circle_with_radius(10)  
    return 3.14 * (10 ** 2)
```

```
area_of_circle_with_radius(10)
```

Variable Scope

- Outside functions = Global variable
- Inside functions = Local variable

Global

```
G = 6.67e-11 #kg-1 m3 s-2
```

```
pi = 3.14
```

```
def Luminosity(radius, temp):
```

```
    sigma_sb = 5.67e-8 #watt m-2 K-4
```

```
    L = 4*pi*radius**2 * sigma_sb * temp**4
```

```
    return L
```

Local



Time for a Coding Demo