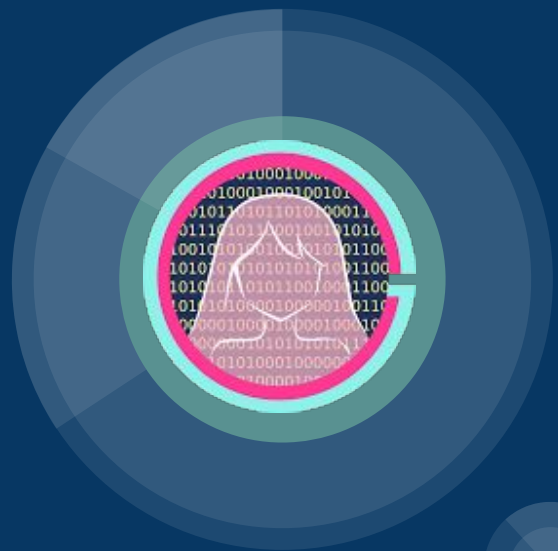


Python Functions

Code Girls 2020-21





Review of Functions

- Functions
 - A block of reusable code to perform a single action
- There are built in functions
- We can also create our own functions!



Define a Function

- Use the **def** keyword
- Example:

```
def hello();  
  
    print("Hello!")
```



Call a Function

- Use the function name followed by parenthesis
- Example:

```
hello()
```



Arguments

- We can pass information into functions as arguments
 - Any data type
- Arguments inside the parentheses after the function name
- You can add more than one argument
 - Separate with a comma

- Example:

```
def hello(firstName):  
    print("Hello " + firstName)
```

- hello("Iris") will print "Hello Iris"
- hello("Alena") will print "Hello Alena"
- hello("Ashley") will print "Hello Ashley"



Parameters vs. Arguments

- Parameters and arguments can both be used for information that are passed into a function
- For a function:
 - Parameter is the variable listed inside the parentheses
 - Argument is the value that is sent to the function when it is called



Number of Arguments and Arbitrary Arguments

- A function must be called with the correct number of arguments or else you will get an error
- If you do not know how many arguments will be passed into your function, add an asterisk `*` before the parameter name in the function definition
 - It will receive a tuple of arguments and your function will be able to access the items

```
def my_function(*kids):  
    print("The youngest child is " + kids[2])  
  
my_function("Emil", "Tobias", "Linus")
```



Keyword Arguments

- You can also send arguments with the **key = value** syntax so that the order of the arguments does not matter

```
def my_function(child3, child2, child1):  
    print("The youngest child is " + child3)
```

```
my_function(child1 = "Emil", child2 = "Tobias", child3 = "Linus")
```




Arbitrary Keyword Arguments

- Similar to arbitrary arguments but for keyword arguments
- Add two asterisks `**` before the parameter name in the function definition
- The function will receive a dictionary of arguments and can access items

```
def my_function(**kid):  
    print("His last name is " + kid["lname"])  
  
my_function(fname = "Tobias", lname = "Refsnes")
```



Default Parameter Value

- You can give the function a default parameter value so that if we call the function without an argument, it uses the default value

```
def my_function(country = "Norway"):  
    print("I am from " + country)
```

```
my_function("Sweden")  
my_function("India")  
my_function()  
my_function("Brazil")
```

```
I am from Sweden  
I am from India  
I am from Norway  
I am from Brazil
```



Return Values

- To let a function return a value, use the **return** statement

```
def my_function(x):  
    return 5 * x
```

```
print(my_function(3))  
print(my_function(5))  
print(my_function(9))
```



Acknowledgments

Thanks to:

- Ms. Lola Piper for sponsoring our club and guidance
- NCWIT for sponsoring Code Girls