

# Functions

Birmingham-Southern College  
Amber Wagner



# Coding

- \* Why do we code?
- \* What's the big idea behind writing a computer program?



# Functions/Methods

- \* Methods/Functions/Procedures allow programmers to write code once and reuse that code as often as necessary
- \* This is not the same as a loop
- \* You have been using methods that were written for you...can you name one?



# How Methods Work

- \* Let's play Simon Says...



# How Methods Work

- \* What happened in Simon Says?



# Print Method

- \* The print method was defined to write to the screen
- \* It was designed with specific parameters...could be one, could be 10
- \* Each parameter is written to the screen



# Range Function

1. What parameters does it accept?
2. How do we use it?
3. How does it work?



# Range Function

1. What parameters does it accept?

- start value (optional - 0 by default)
- stop value
- step value (optional - 1 by default)



# Range Function

## 2. How do we use it?

- in a for loop: `for i in range(10)`
- in an if statement: `if x in range(10)`



# Range Function

## 3. How does it work?

- The range function actually creates a list of numbers
- Example: `range(0, 50, 5)`

creates the following list

`[0, 5, 10, 15, 20, 25, 30, 35, 40, 45]`



# Writing a Method: Syntax

```
def methodName():
```

```
    #Whatever you want your method to do
```

```
    return
```



# Writing a Method: Syntax

```
def methodName(parameters):
```

```
    #Whatever you want your method to do
```

```
    return
```



# Practice!

**Write a method to draw a  
square with asterisks**



# Example

```
def drawSquare():
```

```
    print('* * * * *')
```

```
    print('*         *')
```

```
    print('*         *')
```

```
    print('*         *')
```

```
    print('* * * * *')
```

```
    return
```



# Example

```
def drawSquare():
```

```
    print('* * * * *')
```

```
    print('*       *')
```

```
    print('*       *')
```

```
    print('*       *')
```

```
    print('* * * * *')
```

```
    return
```

Hmmm...not very generic

- One goal of functions is abstraction
- How can we make this function more generic?



# Example

Now, we can make a square of any length side

```
def drawSquare(sideLength):  
    print('* ' * sideLength)  
    for i in range(sideLength - 2):  
        print('*' + ' ' * (sideLength*2-3) + '*')  
    print('* ' * sideLength)  
    return
```



# Returning a Value

- \* `myList = [3,4,2,6,8,10]`
- \* `print(max(myList))`
- \* What happened? What does this do?



# Writing a Function: Syntax

```
def methodName([parameters]):
```

```
    #Whatever you want your method to do
```

```
    return value
```



# Practice Together

- \* Write a function that takes three values and returns the median.
- \* Rewrite this function so that it accepts a list of numbers and returns the median



# Challenge

An online retailer provides express shipping for many of its items at a rate of \$10.95 for the first item, and \$2.95 for each subsequent item. Write a function that takes the number of items in the order as its only parameter. Return the shipping charge for the order as the function's result. Include a main program that reads the number of items purchased from the user and displays the shipping charge.



# For next class...

- \* Read Zybooks! Make sure you are doing 50% of the activities within the chapters!