More Functions & Scope

Code Appetizer

```
def is_a_holiday(month, date):
    if(date == 7):
        if(month=="December" or month==
"September"):
            return True
    else:
        # There are more holidays, but I am lazy
        return False

this_month = "September"
date = 7

if(is_a_holiday(this_month, date)):
    print "Yay! No class!"
else:
    print "Aww, we have class."
```

1. What does this code print to the screen?

2. Use the following vocabulary words to label all the relevant parts of the code:

variable	integer	function definition
function call	conditional	argument
parameter	string	comment
boolean expression		

Hackbright Academy

Code Appetizer - Solution

```
def is_a_holiday(month, date):
    if(date == 7):
        if(month == "December" or month ==
"September"):
            return True
    else:
        # There are more holidays, but I am lazy
        return False

this_month = "September"
date = 7

if(is_a_holiday(this_month, date)):
    print "Yay! No class!"
else:
    print "Aww, we have class."
```

1. What does this code print to the screen?

   "Yay! No class!"

2. Use the following vocabulary words to label all the relevant parts of the code:

variable	integer	function definition
function call	conditional	argument
parameter	string	comment
boolean expression		

Hackbright Academy

Code Appetizer - Solution

```
def is_a_holiday(month, date):
    if(date == 7):
        if(month == "December" or month ==
"September"):
            return True
    else:
        # There are more holidays, but I am lazy
        return False

this_month = "September"
date = 7

if(is_a_holiday(this_month, date)):
    print "Yay! No class!"
else:
    print "Aww, we have class."
```

1. What does this code print to the screen?

   "Yay! No class!"

2. Use the following vocabulary words to label all the relevant parts of the code:

variable	integer	function definition
function call	conditional	argument
parameter	string	comment
boolean expression		

Hackbright Academy

Code Appetizer - Solution

```
def is_a_holiday(month, date):
    if(date == 7):
        if(month == "December" or month ==
"September"):
            return True
    else:
        # There are more holidays, but I am lazy
        return False

this_month = "September"
date = 7

if(is_a_holiday(this_month, date)):
    print "Yay! No class!"
else:
    print "Aww, we have class."
```

1. What does this code print to the screen?

   "Yay! No class!"

2. Use the following vocabulary words to label all the relevant parts of the code:

variable	integer	function definition
function call	conditional	argument
parameter	string	comment
boolean expression		

Hackbright Academy

Code Appetizer - Solution

```
def is_a_holiday(month, date):
    if(date == 7):
        if(month == "December" or month ==
"September"):
            return True
    else:
        # There are more holidays, but I am lazy
        return False

this_month = "September"
date = 7

if(is_a_holiday(this_month, date)):
    print "Yay! No class!"
else:
    print "Aww, we have class."
```

1. What does this code print to the screen?

   "Yay! No class!"

2. Use the following vocabulary words to label all the relevant parts of the code:

   variable, integer, **function definition**, function call, conditional, argument, parameter, string, comment, boolean expression

Hackbright Academy

Code Appetizer - Solution

```
def is_a_holiday(month, date):
    if(date == 7):
        if(month == "December" or month ==
"September"):
            return True
    else:
        # There are more holidays, but I am lazy
        return False

this_month = "September"
date = 7

if(is_a_holiday(this_month, date)):
    print "Yay! No class!"
else:
    print "Aww, we have class."
```

1. What does this code print to the screen?

   "Yay! No class!"

2. Use the following vocabulary words to label all the relevant parts of the code:

variable	integer	function definition
function call	conditional	argument
parameter	string	comment
boolean expression		

Hackbright Academy

Code Appetizer - Solution

```
def is_a_holiday(month, date):
    if(date == 7):
        if(month == "December" or month ==
"September"):
            return True
    else:
        # There are more holidays, but I am lazy
        return False

this_month = "September"
date = 7

if(is_a_holiday(this_month, date)):
    print "Yay! No class!"
else:
    print "Aww, we have class."
```

1. What does this code print to the screen?

   "Yay! No class!"

2. Use the following vocabulary words to label all the relevant parts of the code:

variable	integer	function definition
function call	**conditional**	argument
parameter	string	comment
boolean expression		

Hackbright Academy

Code Appetizer - Solution

```
def is_a_holiday(month, date):
    if(date == 7):
        if(month == "December" or month ==
"September"):
            return True
    else:
        # There are more holidays, but I am lazy
        return False

this_month = "September"
date = 7

if(is_a_holiday(this_month, date)):
    print "Yay! No class!"
else:
    print "Aww, we have class."
```

1. What does this code print to the screen?

   "Yay! No class!"

2. Use the following vocabulary words to label all the relevant parts of the code:

variable	integer	function definition
function call	conditional	**argument**
parameter	string	comment
boolean expression		

Hackbright Academy

Code Appetizer - Solution

```
def is_a_holiday(month, date):
    if(date == 7):
        if(month == "December" or month ==
"September"):
            return True
    else:
        # There are more holidays, but I am lazy
        return False

this_month = "September"
date = 7

if(is_a_holiday(this_month, date)):
    print "Yay! No class!"
else:
    print "Aww, we have class."
```

1. What does this code print to the screen?

   "Yay! No class!"

2. Use the following vocabulary words to label all the relevant parts of the code:

variable	integer	function definition
function call	conditional	argument
parameter	string	comment
boolean expression		

Hackbright Academy

Code Appetizer - Solution

```
def is_a_holiday(month, date):
    if(date == 7):
        if(month ==  "December" or month ==
"September"):
            return True
    else:
        # There are more holidays, but I am lazy
        return False

this_month = "September"
date = 7

if(is_a_holiday(this_month, date)):
    print "Yay! No class!"
else:
    print "Aww, we have class."
```

1. What does this code print to the screen?

   "Yay! No class!"

2. Use the following vocabulary words to label all the relevant parts of the code:

variable	integer	function definition
function call	conditional	argument
parameter	**string**	comment
boolean expression		

Hackbright Academy

Code Appetizer - Solution

```
def is_a_holiday(month, date):
    if(date == 7):
        if(month == "December" or month ==
"September"):
            return True
    else:
        # There are more holidays, but I am lazy
        return False

this_month = "September"
date = 7

if(is_a_holiday(this_month, date)):
    print "Yay! No class!"
else:
    print "Aww, we have class."
```

1. What does this code print to the screen?

   "Yay! No class!"

2. Use the following vocabulary words to label all the relevant parts of the code:

   variable, integer, function definition, function call, conditional, argument, parameter, string, **comment**, boolean expression

Hackbright Academy

Code Appetizer - Solution

```
def is_a_holiday(month, date):
    if(date == 7):
        if(month == "December" or month ==
"September"):
            return True
    else:
        # There are more holidays, but I am lazy
        return False

this_month = "September"
date = 7

if(is_a_holiday(this_month, date)):
    print "Yay! No class!"
else:
    print "Aww, we have class."
```

1. What does this code print to the screen?

   "Yay! No class!"

2. Use the following vocabulary words to label all the relevant parts of the code:

variable	integer	function definition
function call	conditional	argument
parameter	string	comment
boolean expression		

Hackbright Academy

Agenda

Code Appetizer: 6:30 - 6:45

Scope Lecture: 6:45 - 7:15

Functions Exercise 2: 7:15 - 8:00

Break: 8:00 - 8:15

Conditionals Exercise 2: 8:15 - 8:55

Hackbright Bart Simulator: 8:55-9:25

Exit Ticket: 9:25 - 9:30

Scope

Scope

Rules that help the computer decide which variable you are referring to in your code.

Especially useful when you have multiple variables with the same name in your program.

Scope

LEGB rule:

Local, **E**nclosed, **G**lobal, **B**uilt-ins

In this class, we'll only deal with Local and Global scopes.

Scope

- Any variables defined **inside** of function definitions are in the *local scope*.
- Variables defined **outside** of functions are in the *global scope*.

```
num = 8
def double(num):
    num = 1
    doubled = num * 2
    return doubled
print double(num)
```

Hackbright Academy

Scope

When the computer reaches a variable, it:

1. Looks for the variable's definition within local scope.
2. If it's not found, then it looks in global scope.

```
num = 8
def double(num):
    num = 1
    doubled = num * 2
    return doubled
print double(num)
```

There is no local scope, so just looks for `num` in global scope

Hackbright Academy

Scope

When the computer reaches a variable, it:

1. Looks for the variable's definition within local scope.
2. If it's not found, then it looks in global scope.

```
num = 8
def double(num):
    num = 1
    doubled = num * 2
    return doubled
print double(num)
```

Finds the definition of num in global scope.

There is no local scope, so just looks for `num` in global scope

Scope

When the computer reaches a variable, it:

1. Looks for the variable's definition within local scope.
2. If it's not found, then it looks in global scope.

```
num = 8
def double(num):
    num = 1
    doubled = num * 2
    return doubled
print double(num)
```

Looks for num in local scope

Hackbright Academy

Scope

When the computer reaches a variable, it:

1. Looks for the variable's definition within local scope.
2. If it's not found, then it looks in global scope.

```
num = 8
def double(num):
    num = 1
    doubled = num * 2
    return doubled
print double(num)
```

Finds the most recent definition of num in local scope

Looks for num in local scope

Hackbright Academy

Scope

To keep track of all the scopes, variables, and values, it’s good to draw a table.

Scope Example - Flow of Execution

⇒
```
num = 8
def double(num):
    num = 1
    doubled = num * 2
    return doubled
print double(num)
```

Global scope:

```
num = 8
```

Hackbright Academy

Scope Example - Flow of Execution

```
    num = 8
⇒   def double(num):
        num = 1
        doubled = num * 2
        return doubled
    print double(num)
```

Global scope:

```
num = 8
```

Hackbright Academy

Scope Example - Flow of Execution

```
    num = 8
    def double(num):
        num = 1
        doubled = num * 2
        return doubled
⇒   print double(num)
```

Global scope:

```
num = 8
```

Hackbright Academy

Scope Example - Flow of Execution

```
    num = 8
⇒   def double(num):
        num = 1
        doubled = num * 2
        return doubled
⇒   print double(num)
```

`double` (local) scope:

```
num = 8
```

Global scope:

```
num = 8
```

Hackbright Academy

Scope Example - Flow of Execution

```
    num = 8
    def double(num):
⇒       num = 1
        doubled = num * 2
        return doubled
⇒   print double(num)
```

`double` (local) scope:

```
num = 1
```

Global scope:

```
num = 8
```

Hackbright Academy

Scope Example - Flow of Execution

```
    num = 8
    def double(num):
        num = 1
⇒       doubled = num * 2
        return doubled
⇒   print double(num)
```

`double` (local) scope:

```
num = 1
double = 2
```

Global scope:

```
num = 8
```

Scope Example - Flow of Execution

```
    num = 8
    def double(num):
        num = 1
        doubled = num * 2
⇒       return doubled
⇒   print double(num)
```

`double` (local) scope:

```
num = 1
double = 2
```

Global scope:

```
num = 8
```

Scope Example - Flow of Execution

```
    num = 8
    def double(num):
        num = 1
        doubled = num * 2
        return doubled
⇒   print double(num)
```

`double` (local) scope:

```
num = 1
double = 2
```

Global scope:

```
num = 8
```

Code Appetizer - Flow of Execution

```
⇒   def is_a_holiday(month, date):
        if(date == 7):
            if(month=="December" or month=="September"):
                return True
        else:
            # There are more holidays, but I am lazy
            return False

    this_month = "September"
    date = 7

    if(is_a_holiday(this_month, date)):
        print "Yay! No class!"
    else:
        print "Aww, we have class."
```

Global scope:

Code Appetizer - Flow of Execution

```
def is_a_holiday(month, date):
    if(date == 7):
        if(month=="December" or month=="September"):
            return True
    else:
        # There are more holidays, but I am lazy
        return False

⇒   this_month = "September"
    date = 7

    if(is_a_holiday(this_month, date)):
        print "Yay! No class!"
    else:
        print "Aww, we have class."
```

Global scope:

```
this_month =
"September"
```

Hackbright Academy

Code Appetizer - Flow of Execution

```
def is_a_holiday(month, date):
    if(date == 7):
        if(month=="December" or month=="September"):
            return True
    else:
        # There are more holidays, but I am lazy
        return False

this_month = "September"
⇒ date = 7

if(is_a_holiday(this_month, date)):
    print "Yay! No class!"
else:
    print "Aww, we have class."
```

Global scope:

```
this_month =
"September"
date = 7
```

Hackbright Academy

Code Appetizer - Flow of Execution

```
def is_a_holiday(month, date):
    if(date == 7):
        if(month=="December" or month=="September"):
            return True
    else:
        # There are more holidays, but I am lazy
        return False

this_month = "September"
date = 7

⇒ if(is_a_holiday(this_month, date)):
    print "Yay! No class!"
else:
    print "Aww, we have class."
```

Global scope:

```
this_month =
"September"
date = 7
```

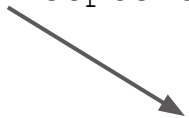
Hackbright Academy

Code Appetizer - Flow of Execution

```
def is_a_holiday(month, date):
    if(date == 7):
        if(month=="December" or month=="September"):
            return True
    else:
        # There are more holidays, but I am lazy
        return False

this_month = "September"
date = 7

⇒ if(is_a_holiday(this_month, date)):
    print "Yay! No class!"
else:
    print "Aww, we have class."
```



Global scope:

```
this_month =
"September"
date = 7
```

Hackbright Academy

Code Appetizer - Flow of Execution

```
def is_a_holiday(month, date):
    if(date == 7):
        if(month=="December" or month=="September"):
            return True
    else:
        # There are more holidays, but I am lazy
        return False

this_month = "September"
date = 7

                "September"
⇒   if(is_a_holiday(this_month, date)):
    print "Yay! No class!"
else:
    print "Aww, we have class."
```

Global scope:

```
this_month =
"September"
date = 7
```

Hackbright Academy

Code Appetizer - Flow of Execution

```
def is_a_holiday(month, date):
    if(date == 7):
        if(month=="December" or month=="September"):
            return True
    else:
        # There are more holidays, but I am lazy
        return False

this_month = "September"
date = 7

⇒ if(is_a_holiday(this_month, date)):
    print "Yay! No class!"
else:
    print "Aww, we have class."
```

"September"

Global scope:

```
this_month =
"September"
date = 7
```

Hackbright Academy

Code Appetizer - Flow of Execution

```
def is_a_holiday(month, date):
    if(date == 7):
        if(month=="December" or month=="September"):
            return True
    else:
        # There are more holidays, but I am lazy
        return False

this_month = "September"
date = 7

                "September"     7
⇒   if(is_a_holiday(this_month, date)):
    print "Yay! No class!"
else:
    print "Aww, we have class."
```

Global scope:

```
this_month =
"September"
date = 7
```

Hackbright Academy

Code Appetizer - Flow of Execution

```
⇒   def is_a_holiday(month, date):
        if(date == 7):
            if(month=="December" or month=="September"):
                return True
        else:
            # There are more holidays, but I am lazy
            return False

    this_month = "September"
    date = 7

                  "September"     7
⇒   if(is_a_holiday(this_month, date)):
        print "Yay! No class!"
    else:
        print "Aww, we have class."
```

is_a_holiday scope:
month = "September"
date = 7

Global scope:
this_month = "September"
date = 7

Hackbright Academy

Code Appetizer - Flow of Execution

```
    def is_a_holiday(month, date):
⇒       if(date == 7):
            if(month=="December" or month=="September"):
                return True
        else:
            # There are more holidays, but I am lazy
            return False

    this_month = "September"
    date = 7

                    "September"    7
⇒   if(is_a_holiday(this_month, date)):
        print "Yay! No class!"
    else:
        print "Aww, we have class."
```

is_a_holiday scope:
month = "September"
date = 7

Global scope:
this_month = "September"
date = 7

Hackbright Academy

Code Appetizer - Flow of Execution

```
    def is_a_holiday(month, date):
⇒       if(date == 7): True
            if(month=="December" or month=="September"):
                return True
        else:
            # There are more holidays, but I am lazy
            return False

    this_month = "September"
    date = 7

                    "September"    7
⇒   if(is_a_holiday(this_month, date)):
        print "Yay! No class!"
    else:
        print "Aww, we have class."
```

is_a_holiday scope:
month = "September"
date = 7

Global scope:
this_month = "September"
date = 7

Hackbright Academy

Code Appetizer - Flow of Execution

```
    def is_a_holiday(month, date):
        if(date == 7): True
⇒           if(month=="December" or month=="September"):
                return True
        else:
            # There are more holidays, but I am lazy
            return False

    this_month = "September"
    date = 7

                    "September"    7
⇒   if(is_a_holiday(this_month, date)):
        print "Yay! No class!"
    else:
        print "Aww, we have class."
```

is_a_holiday scope:
month = "September"
date = 7

Global scope:
this_month = "September"
date = 7

Hackbright Academy

Code Appetizer - Flow of Execution

```
    def is_a_holiday(month, date):
        if(date == 7): True   False
⇒           if(month=="December" or month=="September"):
                return True
        else:
            # There are more holidays, but I am lazy
            return False

    this_month = "September"
    date = 7

                  "September"    7
⇒   if(is_a_holiday(this_month, date)):
        print "Yay! No class!"
    else:
        print "Aww, we have class."
```

is_a_holiday scope:
month = "September"
date = 7

Global scope:
this_month = "September"
date = 7

Hackbright Academy

Code Appetizer - Flow of Execution

```
    def is_a_holiday(month, date):
        if(date == 7): True   False
⇒           if(month=="December" or month=="September"):
                return True
        else:
            # There are more holidays, but I am lazy
            return False

    this_month = "September"
    date = 7

                    "September"    7
⇒   if(is_a_holiday(this_month, date)):
        print "Yay! No class!"
    else:
        print "Aww, we have class."
```

is_a_holiday scope:
month = "September"
date = 7

Global scope:
this_month = "September"
date = 7

Code Appetizer - Flow of Execution

```
    def is_a_holiday(month, date):
        if(date == 7):  True   False          True
⇒           if(month=="December" or month=="September"):
                return True
        else:
            # There are more holidays, but I am lazy
            return False

    this_month = "September"
    date = 7

                    "September"    7
⇒   if(is_a_holiday(this_month, date)):
        print "Yay! No class!"
    else:
        print "Aww, we have class."
```

is_a_holiday scope:
month = "September"
date = 7

Global scope:
this_month = "September"
date = 7

Code Appetizer - Flow of Execution

```
    def is_a_holiday(month, date):
        if(date == 7):  True   False          True
⇒           if(month=="December" or month=="September"):
                return True
        else:
            # There are more holidays, but I am lazy
            return False

    this_month = "September"
    date = 7

                  "September"    7
⇒   if(is_a_holiday(this_month, date)):
        print "Yay! No class!"
    else:
        print "Aww, we have class."
```

is_a_holiday scope:
month = "September"
date = 7

Global scope:
this_month = "September"
date = 7

Code Appetizer - Flow of Execution

```
    def is_a_holiday(month, date):
        if(date == 7): True  False  True  True
⇒           if(month=="December" or month=="September"):
                return True
        else:
            # There are more holidays, but I am lazy
            return False

    this_month = "September"
    date = 7

                    "September"    7
⇒   if(is_a_holiday(this_month, date)):
        print "Yay! No class!"
    else:
        print "Aww, we have class."
```

is_a_holiday scope:
month = "September"
date = 7

Global scope:
this_month = "September"
date = 7

Code Appetizer - Flow of Execution

```
    def is_a_holiday(month, date):
        if(date == 7):  True
            if(month=="December" or month=="September"):
                                          True
⇒               return True
        else:
            # There are more holidays, but I am lazy
            return False

    this_month = "September"
    date = 7

                    "September"     7
⇒   if(is_a_holiday(this_month, date)):
        print "Yay! No class!"
    else:
        print "Aww, we have class."
```

is_a_holiday scope:
month = "September"
date = 7

Global scope:
this_month = "September"
date = 7

Code Appetizer - Flow of Execution

```
def is_a_holiday(month, date):
    if(date == 7):  True
        if(month=="December" or month=="September"):
            return True

    else:
        # There are more holidays, but I am lazy
        return False

this_month = "September"
date = 7

⇒ if(is_a_holiday(this_month, date)):
    print "Yay! No class!"
else:
    print "Aww, we have class."
```

True

True

is_a_holiday scope:
month = "September"
date = 7

Global scope:
this_month = "September"
date = 7

Hackbright Academy

Code Appetizer - Flow of Execution

```
def is_a_holiday(month, date):
    if(date == 7):  True
                                    True
        if(month=="December" or month=="September"):
            return True
    else:
        # There are more holidays, but I am lazy
        return False

this_month = "September"
date = 7

                    True
if(is_a_holiday(this_month, date)):
⇒       print "Yay! No class!"
else:
    print "Aww, we have class."
```

is_a_holiday scope:
month = "September"
date = 7

Global scope:
this_month = "September"
date = 7

Hackbright Academy

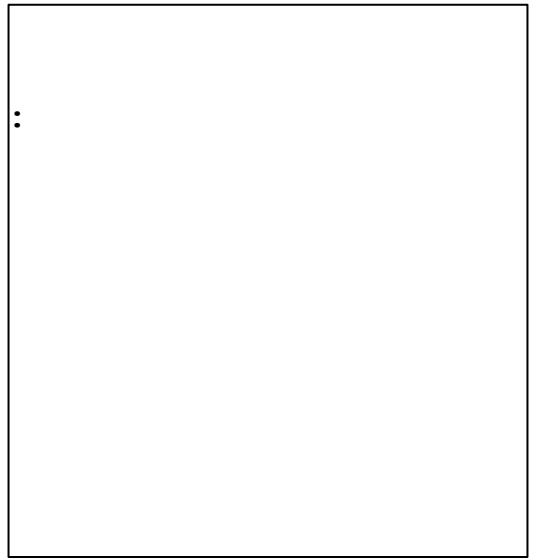
Code Appetizer - Flow of Execution

```
def is_a_holiday(month, date):
    if(date == 7):          True
        if(month=="December" or month=="September"):
            return True                    True
    else:
        # There are more holidays, but I am lazy
        return False

this_month = "September"
date = 7

                    True
if(is_a_holiday(this_month, date)):
    print "Yay! No class!"
else:
    print "Aww, we have class."
```

⇒ **Program finished executing. It exits (quits).**
All variables are destroyed.



Hackbright Academy