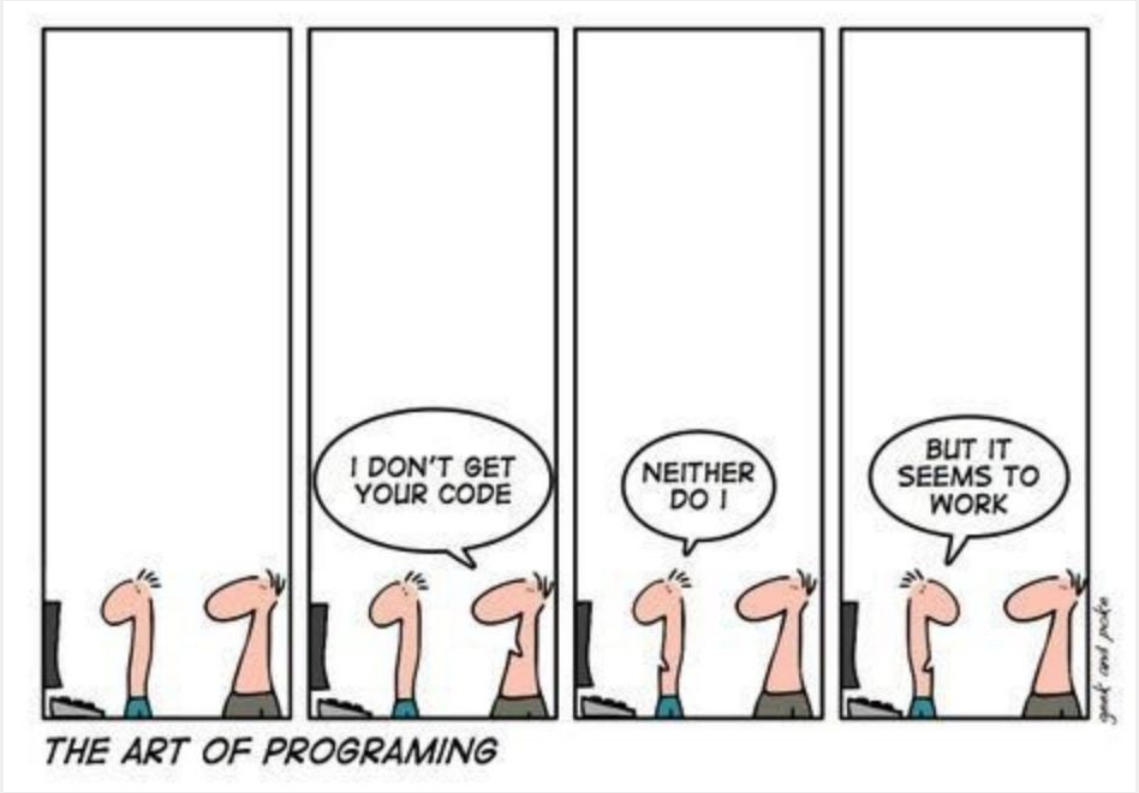

CS 110 Functions

Adriana Picoral
(she/her/hers)

What is redundant?

```
print('--- Weekend Planner ---')
print('--- Friday Tasks ---')
friday_tasks = ''
task = ''
while task != 'sleep':
    task = input('Next task for Friday: ')
    friday_tasks += ' * ' + task + '\n'
print('--- Saturday Tasks ---')
saturday_tasks = ''
task = ''
while task != 'sleep':
    task = input('Next task for Saturday: ')
    saturday_tasks += ' * ' + task + '\n'
print('--- Sunday Tasks ---')
sunday_tasks = ''
task = ''
while task != 'sleep':
    task = input('Next task for Sunday: ')
    sunday_tasks += ' * ' + task + '\n'
```

```
print('+-----------------------+')
print('+--- Weekend Summary ---+')
print('+-----------------------+')
print('+--------- Friday --------+')
print(friday_tasks)
print('+--------- Saturday --------+')
print(saturday_tasks)
print('+--------- Sunday --------+')
print(sunday_tasks)
```

What is redundant?

```
print('--- Weekend Planner ---')
print('--- Friday Tasks ---')
friday_tasks = ''
task = ''
while task != 'sleep':
    task = input('Next task for Friday: ')
    friday_tasks += ' * ' + task + '\n'
print('--- Saturday Tasks ---')
saturday_tasks = ''
task = ''
while task != 'sleep':
    task = input('Next task for Saturday: ')
    saturday_tasks += ' * ' + task + '\n'
print('--- Sunday Tasks ---')
sunday_tasks = ''
task = ''
while task != 'sleep':
    task = input('Next task for Sunday: ')
    sunday_tasks += ' * ' + task + '\n'
```

```
print('+-----------------------+')
print('+--- Weekend Summary ---+')
print('+-----------------------+')
print('+--------- Friday -------+')
print(friday_tasks)
print('+--------- Saturday -------+')
print(saturday_tasks)
print('+--------- Sunday -------+')
print(sunday_tasks)
```

What is different?

```python
print('--- Weekend Planner ---')
print('--- Friday Tasks ---')
friday_tasks = ''
task = ''
while task != 'sleep':
    task = input('Next task for Friday: ')
    friday_tasks += ' * ' + task + '\n'
print('--- Saturday Tasks ---')
saturday_tasks = ''
task = ''
while task != 'sleep':
    task = input('Next task for Saturday: ')
    saturday_tasks += ' * ' + task + '\n'
print('--- Sunday Tasks ---')
sunday_tasks = ''
task = ''
while task != 'sleep':
    task = input('Next task for Sunday: ')
    sunday_tasks += ' * ' + task + '\n'
```

```python
print('+------------------------+')
print('+--- Weekend Summary ---+')
print('+------------------------+')
print('+--------- Friday --------+')
print(friday_tasks)
print('+--------- Saturday --------+')
print(saturday_tasks)
print('+--------- Sunday --------+')
print(sunday_tasks)
```

Activity

Write a function to get rid of redundancy

```
print('--- Friday Tasks ---')
friday_tasks = ''
task = ''
while task != 'sleep':
    task = input('Next task for Friday: ')
    friday_tasks += ' * ' + task + '\n'

print('--- Saturday Tasks ---')
saturday_tasks = ''
task = ''
while task != 'sleep':
    task = input('Next task for Saturday: ')
    saturday_tasks += ' * ' + task + '\n'

print('--- Sunday Tasks ---')
sunday_tasks = ''
task = ''
while task != 'sleep':
    task = input('Next task for Sunday: ')
    sunday_tasks += ' * ' + task + '\n'
```

```
tasks = ''

def get_day_tasks(day):
    # What should go here ???


get_day_tasks('Friday')
get_day_tasks('Saturday')
get_day_tasks('Sunday')
```

```python
tasks = '+-----------------------+\n+--- Weekend Summary ---+\n+-----------------------+\n'

def get_day_tasks(day):
    print('--- ' + day + ' Tasks ---')









get_day_tasks('Friday')
get_day_tasks('Saturday')
get_day_tasks('Sunday')
print(tasks)
```

```python
tasks = '+-----------------------+\n+--- Weekend Summary ---+\n+-----------------------+\n'

def get_day_tasks(day):
    print('--- ' + day + ' Tasks ---')
    tasks += '+-------- ' + day + ' -------+\n'




get_day_tasks('Friday')
get_day_tasks('Saturday')
get_day_tasks('Sunday')
print(tasks)
```

```python
tasks = '+----------------------+\n+--- Weekend Summary ---+\n+----------------------+\n'

def get_day_tasks(day):
    print('--- ' + day + ' Tasks ---')
    tasks += '+--------- ' + day + ' --------+\n'
    task = ''
    while task != 'sleep':
        task = input('Next task for ' + day + ': ')
        tasks += ' * ' + task + '\n'


get_day_tasks('Friday')
get_day_tasks('Saturday')
get_day_tasks('Sunday')
print(tasks)
```

```python
tasks = '+-----------------------+\n+--- Weekend Summary ---+\n+-----------------------+\n'

def get_day_tasks(day):
    global tasks
    print('--- ' + day + ' Tasks ---')
    tasks += '+--------- ' + day + ' --------+\n'
    task = ''
    while task != 'sleep':
        task = input('Next task for ' + day + ': ')
        tasks += ' * ' + task + '\n'

get_day_tasks('Friday')
get_day_tasks('Saturday')
get_day_tasks('Sunday')
print(tasks)
```

```python
def get_day_tasks(day):
    print('--- ' + day + ' Tasks ---')
    tasks = '+-------- ' + day + ' -------+\n'
    task = ''
    while task != 'sleep':
        task = input('Next task for ' + day + ': ')
        tasks += ' * ' + task + '\n'
    return tasks

def main():
    tasks_fr = get_day_tasks('Friday')
    tasks_sa = get_day_tasks('Saturday')
    tasks_su = get_day_tasks('Sunday')
    print('+-----------------------+\n+--- Weekend Summary ---+\n+-----------------------+\n')
    print(tasks_fr)
    print(tasks_sa)
    print(tasks_su)

main()
```

Returning a value

- Using we can send a value to a function using **arguments** and **parameter variables**.
- We can also **return** values from a function using the **return** statement
- It is often useful to have a function yield a particular value

```
def function_name():
    statementA
    . . .
    statementN

statement . . .

function_name()

statements . . .
```

```
def function_name():
    statementA
    . . .
    return n

statement . . .

var = function_name()

statements . . .
```

```
def function_name():
    statementA
    . . .
    return

statement . . .

function_name()

statements . . .
```

```
def function_name():
    statementA
    if …:
        return
    statementY

statement . . .

function_name()

statements . . .
```

```
def categorize(height):
    if height > 70:
        return "tall"
    else:
        return "short"

statements . . .

category_1 = categorize(75)
category_2 = categorize(65)

statements . . .
```

Activity

What would this print?

```
def repeat(content, times):
    to_return = content * times
    return to_return

result = repeat('110', 5)
print(result)
```

What would this print?

```
def repeat(content, times):
    to_return = ''
    i = 0
    while i < times:
        to_return += content
        i += 1
    return to_return

result = repeat('110', 5)
print(result)
```

Activity

The pythagorean theorem

https://en.wikipedia.org/wiki/Pythagorean_theorem

$$a^2 + b^2 = c^2$$

$$c = \sqrt{a^2 + b^2}.$$

Activity

The pythagorean theorem

- Write a function that accepts two ints as parameters
- These represent the length of the two non-hypotenuse sides
- Returns the length of the hypotenuse

$$a^2 + b^2 = c^2$$

$$c = \sqrt{a^2 + b^2}.$$

```
# return 5.0
pythagorean(3, 4)
# return 14.142135623730951
pythagorean (10, 10)
```

Implement the pythagorean function

```python
def pythagorean(a, b):
    c_squared = (a**2 + b**2)
    c = (c_squared)**0.5
    return c
```

Implement the pythagorean function

```python
def pythagorean(a, b):
    return (a**2 + b**2)**0.5
```

```python
def pythagorean(a, b):
    '''
     Calculates the length of c (the hypotenuse) of a right triangle using
     the pythagorean theorem.
     a and b: The length of the sides of a right-triangle that are adjacent
              to the right-angle.
     Returns an integer that is the calculated length of side c.
     '''
    c_squared = (a**2 + b**2)
    c = (c_squared)**0.5
    return c

def main():
    a_value = float(input('Enter a value: '))
    b_value = float(input('Enter b value: '))
    result = pythagorean(a_value, b_value)
    print(result)

main()
```