Lab 01: Variables & Functions, Control

> Adapted form cs61a of UC Berkeley.

Starter Files

Get your starter file by cloning the repository: https://github.com/JacyCui/sicp-lab01.git

```
git clone https://github.com/JacyCui/sicp-lab01.git
```

`lab01.zip` is the starter file you need, you might need to unzip the file to get the skeleton code.

```
unzip lab01.zip
```

`README.md` is the handout for this homework. `solution` is a probrab solution of the homework. However, I might not give my solution exactly when the homework is posted. You need to finish the task on your own first. If any problem occurs, please make use of the comment section.

Required Questions

What Would Python Display?(Part 1)

Q1: WWPD: Control

> Use Ok to test your knowledge with the following "What Would Python Display?" questions:
>
> ```
> python3 ok -q control -u --local
> ```
>
> **Hint**: Make sure your `while` loop conditions eventually evaluate to a false value, or they'll never stop! Typing `Ctrl-C` will stop infinite loops in the interpreter.

```
>>> def xk(c, d):
...     if c == 4:
...         return 6
...     elif d >= 4:
...         return 6 + 7 + c
...     else:
...         return 25
>>> xk(10, 10)
_____

```

```
>>> xk(10, 6)
_______

>>> xk(4, 6)
_______

>>> xk(0, 0)
_______
```

```
>>> def how_big(x):
...     if x > 10:
...         print('huge')
...     elif x > 5:
...         return 'big'
...     elif x > 0:
...         print('small')
...     else:
...         print("nothin")
>>> how_big(7)
_______

>>> how_big(12)
_______

>>> how_big(1)
_______

>>> how_big(-1)
_______
```

```
>>> n = 3
>>> while n >= 0:
...     n -= 1
...     print(n)
_______
```

```
>>> positive = 28
>>> while positive:
...     print("positive?")
...     positive -= 3
_______
```

```
>>> positive = -9
>>> negative = -12
>>> while negative:
...     if positive:
...         print(negative)
...     positive += 3
...     negative += 3
_____
```

Q2: WWPD: Veritasiness

> Use Ok to test your knowledge with the following "What Would Python Display?" questions:
>
> ```
> python3 ok -q short-circuit -u --local
> ```

```
>>> True and 13
_____

>>> False or 0
_____

>>> not 10
_____

>>> not None
_____
```

```
>>> True and 1 / 0 and False
_____

>>> True or 1 / 0 or False
_____

>>> True and 0
_____

>>> False or 1
_____

>>> 1 and 3 and 6 and 10 and 15
_____

>>> -1 and 1 > 0
_____

>>> 0 or False or 2 or 1 / 0
```

```
20 _______
```

```
1  >>> not 0
2  _______
3
4  >>> (1 + 1) and 1
5  _______
6
7  >>> 1/0 or True
8  _______
9
10 >>> (True or False) and False
11 _______
```

Q3: Debugging Quiz!

The following is a quick quiz on different debugging techniques you should use in this class. You should refer to `Debugging.md` to answer the questions!

Run the following to run the quiz.

```
1 python3 ok -q debugging-quiz -u --local
```

Coding Practice

Q4: Falling Factorial

Let's write a function `falling`, which is a "falling" factorial that takes two arguments, `n` and `k`, and returns the product of `k` consecutive numbers, starting from `n` and working downwards.

```
1  def falling(n, k):
2      """Compute the falling factorial of n to depth k.
3
4      >>> falling(6, 3)  # 6 * 5 * 4
5      120
6      >>> falling(4, 3)  # 4 * 3 * 2
7      24
8      >>> falling(4, 1)  # 4
9      4
10     >>> falling(4, 0)
11     1
12     """
13     "*** YOUR CODE HERE ***"
```

Use Ok to test your code:

```
python3 ok -q falling --local
```

Q5: Sum Digits

Write a function that takes in a nonnegative integer and sums its digits. (Using floor division and modulo might be helpful here!)

```
def sum_digits(y):
    """Sum all the digits of y.

    >>> sum_digits(10) # 1 + 0 = 1
    1
    >>> sum_digits(4224) # 4 + 2 + 2 + 4 = 12
    12
    >>> sum_digits(1234567890)
    45
    >>> a = sum_digits(123) # make sure that you are using return rather than print
    >>> a
    6
    """
    "*** YOUR CODE HERE ***"
```

Use Ok to test your code:

```
python3 ok -q sum_digits --local
```

Extra Practice

> These questions are optional. However, they are **great practice** for future assignments and projects. Attempting these questions is valuable in helping cement your knowledge of course concepts, and it's fun!

What Would Python Display? (Part 2)

Q6: WWPD: What If?

> Use Ok to test your knowledge with the following "What Would Python Display?" questions:
>
> ```
> python3 ok -q if-statements -u --local
> ```
>
> **Hint**: `print` (unlike `return`) does *not* cause the function to exit!

```
>>> def ab(c, d):
...     if c > 5:
...         print(c)
...     elif c > 7:
...         print(d)
...     print('foo')
>>> ab(10, 20)
_____
```

```
>>> def bake(cake, make):
...     if cake == 0:
...         cake = cake + 1
...         print(cake)
...     if cake == 1:
...         print(make)
...     else:
...         return cake
...     return make
>>> bake(0, 29)
_____

>>> bake(1, "mashed potatoes")
_____
```

More Coding Practice

Q7: Double Eights

Write a function that takes in a number and determines if the digits contain two adjacent

```
def double_eights(n):
    """Return true if n has two eights in a row.
    >>> double_eights(8)
    False
    >>> double_eights(88)
    True
    >>> double_eights(2882)
    True
    >>> double_eights(880088)
    True
    >>> double_eights(12345)
    False
    >>> double_eights(80808080)
    False
    """
    "*** YOUR CODE HERE ***"
```

Use Ok to test your code:

```
python3 ok -q double_eights --local
```