Programming Layer

AP CSP @ SouthLake Christian Academy

Basic Terminology

- A program is a collection of program statements that performs a specific task when run by a computer. A program is often referred to as software.
- A code segment is a collection of program statements that is part of a program.

Program function is how softare accomplishes a task

Program purpose is why software accomplishes a task

We will be programming in Python

- Python files end in `.py`
- Python allows for programs to work basically the same on any machine

Pseudocode

- **pseudocode**: "false code" → an informal tool that uses English to represent a set of *strict* instructions

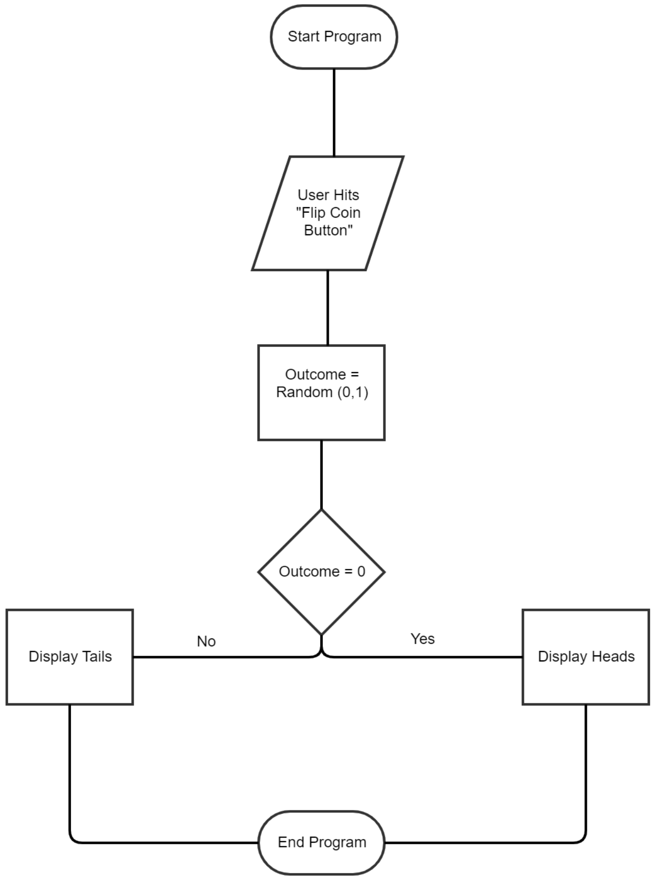
```
0. Do first step
1. Do second step
2. Do next step...
```

Pseudocode for: Brushing Teeth

- We have instructions that occur in *sequence*
- Some of these lines start with verbs, or actions. We'll start calling these *functions* or *procedures*
- We also have questions that lead to different paths, like forks in the road, which we'll call *conditions*:
- And the answers to questions that decide where we go are called *Boolean expressions*, which eventually result to a value of true or false
- Finally, we have words that lead to cycles, where we can repeat parts of our program, called *loops* or *iterations*:

Flowcharts

Another useful tool for visualizing a set of instructions



Comments

are not part of the program; the Python interpreter will completely ignore lines that start with `#` or text enclosed in `"""`

```
"""
This text is ignored. Use comments to describe your program!

I can write multi-line comments!
"""

# This line is also ignored
```

Variables

a *variable* is a piece of memory that holds some value

NOTE: x = 1 means "assign x the value of 1" NOT "x equals 1"

Variable Operations

Operator	Example
+	y = x + 1
-	y = x - 1
*	y = x * 2
**	y = x ** 2
/	y = x / 2
//	y = x // 2
%	y = x % 10

`print`

displays content to console

```
print("Hello, World!")
>> Hello, World!
>>
```

specify the line ending with end:

```
print("Hello, World!", end="a")
>> Hello, World!a
```

`\n` is a special character that specifies newlines

`\t` is a special character that specifies tabs

by default, `end="\n"`

`input`

asks the user for some input, returns a string of characters

will wait until user types something and presses Enter/Return

```
x = input("What is your name?")
```

Functions

abstractions that allow us to group related code statements

```
def myFunction():
    print("Yay")
```

Arguments and Returns

parameters or *arguments* are inputs to functions

```
def myFunction(msg):
    print("I was given:")
    print(msg)
```

a function can *return* a value to the function caller:

```
def addOne(x):
    return x + 1
```