Cheatsheets / **Learn How to Code**

Basics of Programming II

Loop Definition

In programming, a *loop* is a programming structure that repeats a set of instructions until a specified condition is met. *Loops* are commonly used in programming because, compared to repeated lines of code, they save time, reduce error, and are easy to read.
A loop to play a sound may look like this:

```
UNTIL 4 sounds have been played:
    Play a sound
```

Conditional Control

Conditional statements or *conditional* control structures allow a program to have different behaviors depending on certain *conditions* being met.
Intuitively, this mimics the way humans make simple decisions and act upon them. For example, reasoning about whether to go outside might look like:

- Condition: *Is it raining outside?*
    - If it is raining outside, then *bring an umbrella.*
    - Otherwise, *do not bring an umbrella.*

We could keep adding clauses to make our reasoning more sophisticated, such as “If it is sunny, then wear sunscreen”.

Control Flow

In programming, *control flow* is the order in which statements and instructions are executed. Programmers are able to change a program's *control flow* using *control structures* such as conditionals.
Being able to alter a program's *control flow* is powerful, as it lets us adapt a running program's behavior depending on the state of the program. For example, suppose a user is using a banking application and wants to withdraw $500. We certainly want the application to behave differently depending on whether the user has $20 or $1000 in their bank account!

Define Function Definition

A function is *defined* by specifying its instructions, inputs, and name.
For example, a sandwich-making function takes in two inputs representing two ingredients. The definition in psuedocode would look like:

```
function makeSandwich(topping1, topping2) {
  Add bread
  Add topping1
  Add topping2
  Add bread
}
```

The name is `makeSandwich`, the inputs are `topping1` and `topping2`, and the instructions are the four lines each starting with `Add`.

Function Inputs Definitions

"In programming, inputs to functions are known as *parameters* when a function is declared or defined. Parameters are variables that can be used inside the function body. When the function is called, these parameters will have the value of whatever is passed in as *arguments*.
For example, a sandwich-making function has two parameters:

```
function makeSandwich(topping1, topping2) {
  Add bread
  Add topping1
  Add topping2
  Add bread
}
```

We make a ham-and-cheese sandwich with `makeSandwich("ham", "cheese")`. We call the function with the arguments `"ham"` and `"cheese"`. Those will be the values for the `topping1` and `topping2` parameters."

Print Share