

# Day 1

## Objective #1 : Problem Solving

The activity planned will involve a fun intro to logical thinking by playing the game lightbot\*

<http://light-bot.com/>

The desired outcome is that each child will begin to understand that a program is a series of small procedural steps. Each step is simple by itself, but together many simple steps can help to produce a complex program.

\*if each student has access to an iPad/Android tablet I'd recommend Lightbot Jr over the flash based game I link to above. It has a slower ramp to help the students get comfortable with problem solving (also the touch interface works a little better for this type of game format)

<https://itunes.apple.com/us/app/lightbot-jr-4+-coding-puzzles/id858640629?mt=8>

## Objective #2 : Variables

The basic concept will be introduced using the familiar kids game madlibs

<http://www.madlibs.com/>

The idea is to keep the students in a safe place they know and understand (like creating a fun madlib story on paper). To be successful make sure the kids first fill out the top part of the madlib before looking at the story. This will help as we "plug in" each noun/verb to produce a silly story.

At this point be sure to discuss how this is like the computer (when a program runs it will generate something different each time depending on what you input - this is where variables come in). I used the word "placeholder" a great deal because my students understood that concept and with help from the madlib it made perfect sense (feel free to use whatever word helps the students understand the variable idea conceptually).

## Objective #3 : REPL

We will discuss how to "play with variables" using the python REPL

The activity planned will introduce the kids to basic string variables (how to print them / combine them). Also during this activity we will introduce the concept of "keywords" or "reserved" words that python uses, such as "print"

We start with something basic like creating a variable for each students name and age. We then combine them (no space between initially). We then work through how we can combine them using the plus operator. We spend a great deal of time here as it's critical each student understand how variables work for us to build on throughout the week.

Repetition played a big role to help the kids gain experience and feel comfortable learning all the new concepts.

#### **Objective #4 : Intro to types**

Introduction to "types" by showing how numbers are different than strings (previously introduced).

The activity planned will show how we can create a basic calculator with python \*\*along with the basic operators like + - \* /

We replayed the name/age example from the above REPL objective but this time we discussed the difference between numbers and strings (as age could be either depending on what the student decided to use for the value).

We also had a brief introduction to the boolean type showing examples like  $5 < 10$  and  $10 == 12$

## **Day 2**

Warmup with a partner

When the students arrive we will pair them up to work on another problem solving game "lightbot 2" (again if students have access to an iPad/Android tablet I'd recommend lightbot jr instead of the flash based game).

The activity is designed to see how the kids problem solve + collaborate together. (if you find the kids appreciate the competitive edge a "who completes the most levels" you can introduce a challenge of some sort).

Recap from day 1

This activity will recap the REPL / variables / strings vs numbers / from the previous day (repetition was always good for me, so I'll mix it up with a few new examples to keep everyone interested).

#### **Objective #5 : Numbers + Mod Operator**

Using python to solve more complex math problems (including the introduction to mod)

This activity will have us working through a series of problems like "order of operations" / "if a number is even" / "if a number is odd" (using the mod operator)

Depending on the students skill level you might be teaching order of operations for the first time. I used a great deal of fun examples with add/divide/multiply/subtract/mod

#### **Objective #6 : IDLE**

discuss how to write a python program without the REPL (using IDLE)

The activity is to create a python program by saving a new file "example.py" and running it from the command line. This will pave the ground work for the remainder of the week (as we will begin to write larger and larger programs as we move forward).

The main reason we introduce IDLE at this step is because conditionals (our next objective) can cause a lot of confusion/pain when done with the REPL alone. I also found the students really enjoyed using IDLE to create new files / run the program and see results immediately.

Be sure to set a preference in IDLE to "not prompt" when the student clicks run (this will auto save and run the program without asking the student to save over and over again).

### **Objective #7 : Conditionals**

The activity planned will explain what a conditional is and how to use it in python. I have a number of fun exercises that I'll have the students work on individually after we introduce it by example. (if / else / if else using the print statement for starters)

As you work through the examples be aware that most of the students will mixup the spacing / indentation at first. Use this as the opportunity to discuss "scoping" but keep it simple during this initial exercise (more repetition throughout the week will help set this in stone for the students)

This concept is introduced at the end of the day because it will be the most confusing new idea that the students will work through. If you don't have a huge breakthrough initially - remember to keep at it the following day (during the recap).

## **Day 3**

Warmup with a partner (optional)

I'll have 15 exercises for the students to work on in pairs from day 1

Recap from day 2

Everything from this point on will be done using IDLE

This activity will recap the variables / types / simple to complex math problems / conditionals

A great deal of new examples are added here but the concepts are much the same (again using repetition to reinforce and provide experience to help the students feel comfortable over time).

### **Objective #8 : Loops**

This activity will introduce the second most valuable programming concept (how to loop over something)

We start with the basic while loop -using it to print a phrase 5 times in a row. As when conditionals were introduced, be aware that spacing within the loop will need to be called out. I also showed how scoping works by putting a print after the loop and asking the students how many times each print would show up.

## **Objective #9 : Lists**

This activity will introduce the more complex data structure (the list) and how it can hold different types of values / and how to "loop" over it

We quickly introduced the for loop to loop over each item in the list. Be sure to express how the variable is set during each iteration (pulling it's value from the list declared above).

We also showed how to inline a for loop (no variable required).

Several examples showing how to loop over a list of strings / or a list of numbers. One example worth mentioning is the "total up all numbers in a list"

## **Objective #10 : Input from the user**

This activity will introduce the students to `raw_input`, a special function that lets the program "pause" for user input. I have a few fun examples showing how to use this to build more complex programs.

The students really enjoyed this part of the class so be sure to come up with a ton of fun examples and let them experiment. Be sure to cover the variable "type" of `raw_input` starts as a string (you might mention how to convert the string to a number using `int`)

## **Day 4**

Most of the day is spent doing a huge recap of variables / types / conditionals / lists / loops (more repetition to enforce the conditional and looping concepts introduced on day 3)

The push during day 4 is to cover the following concepts (as they will be used directly in day 5 as we build the battleship game)

while loop (several examples that show scoping / how loops work / how to increment a number)

`raw_input` (several examples showing how to ask users for input / pointing out that the program will pause execution until the user types something in and hits enter)

conditionals (several examples using both `if` and `else` / keeping scope in mind throughout)

how to modify an element in a list (we cover zero based indexing yet again as it's confusing)

## **Objective #11 : functions**

This activity will start by showing the `len` function built into python for showing the length of a list. From here we show how to write your own functions and why they are important as your software grows.

We use the experience each student had with functions in lightbot to help them understand how they work in python. We don't use functions in the game but it's great to cover this topic anyway.

## Day 5

Warmup by looking at a gist I prepared that printed “even number” for each even number in a list. The code works but I repeat the for loop / conditional 3 times and ask the students to “shorten” the program.

<https://gist.github.com/toranb/f1153aa9aee57ad942a2>

Recap the entire week using examples from days 1 - 4

### **Objective #12 : Putting it all together**

We will build a very simple battleship game as a class (using everything we've learned throughout the week).

<https://gist.github.com/toranb/74d06bc50055840eaeaf>

### **Objective #13 : Showcase what they learned**

I'll ask that the parents show up 15 minutes early so each student can demo what they've learned (ie- by having the parents play the battleship game along side the student).

We have the parents play the game by launching it from the terminal (instead of IDLE)