

## **Lesson Plan for Activity Python: Mad Libs**

**Subject:** Programming

**Length of Lesson:** 1 hour and 15 minutes

**Concept or Skill Focus:** Python

**Goal:** Learning the basics of Python and how to take the next step in programming with actual coding.

**Objectives/Outcomes:**

- Learn what Python is
- Learn simple Python functions such as printing, user input, variables, and concatenation
- Create a Mad Libs story using the newly learned Python concepts

**Activities and Time Line (Introduction, Middle, Conclusion)**

INTRODUCTION (10 mins):

- Teach the students what Python is (possibly a little bit of history)
- Talk about how Python is used today
- Teach them the functions they will be using in their Mad Libs game and how to use them

MIDDLE (45 mins):

- The pairs of students will work together to make their individual stories

CONCLUSION (20 mins):

- With whatever time is remaining, the kids can go around to other computers to try the other stories.

**Materials**

- Computers
- Python version 3.8 downloadable here: <https://www.python.org/downloads/>
- Python Mad Libs Game reference sheet (below)
- Python Number Guessing Game reference sheet (below)

# Python Mad Libs Game

## Description:

The program will first prompt the user for a series of inputs a la Mad Libs. For example, a singular noun, an adjective, etc. Then, once all the information has been inputted, the program will take that data and place them into a premade story template. You'll need prompts for user input, and to then print out the full story at the end with the input included.

## Concepts You Will Need:

- Strings
- Variables
- Concatenation
- Print

## Code Help:

### Print statement:

```
print("__Whatever text you want to print__")
```

### Variable and user input:

```
variableName = input("__Whatever question you want the user to answer__")
```

### Concatenation:

```
"this is a " + variableName + " and I like it."
```

# Python Number Guessing Game

## Description:

The program will first randomly generate a number unknown to the user. The user needs to guess what that number is. (In other words, the user needs to be able to *input* information.) If the user's guess is wrong, the program should return some sort of indication as to how wrong (e.g. The number is too high or too low). If the user guesses correctly, a positive indication should appear.

## Concepts You Will Need:

- Random function
- Variables
- Integers
- Input/Output
- Print
- While loops
- If/Else statements

## Code Help:

### Print statement:

```
print("__Whatever text you want to print__")
```

### Variable and user input:

```
variableName = input("__Whatever question you want the user to answer__")
```

### Random number generator:

```
import random
```

```
variableName = random.randint(startNumber, totalNumbers)
```

### While loop:

```
While *condition*:
```

### If Else loop:

```
if *condition*:
```

```
elif *condition*:
```

```
...
```

```
else:
```