

The background features abstract geometric lines in teal and dark blue. On the left, several parallel teal lines form a corner-like structure. At the bottom, there are more parallel teal lines, some of which are horizontal and others diagonal. On the right side, a series of parallel teal lines slope upwards from the bottom towards the right edge.

# python

CODE ACADEMY AND PROCESSING

## resources

There are so many useful websites that are created to help you

- CODEACADEMY
- KhanAcademy
- learnPython.org
- Python.org

Python | Codecademy

https://www.codecademy.com/learn/python

codecademy

Upgrade to Pro

Learn

Catalog

Dashboard

Courses

Python

33%

Want more practice and review? Upgrade for the complete experience.

8 Projects

9 Quizzes

1 Final Project

Get Instant Access »

Overview

Syllabus

1

Python Syntax

⌵

2

Strings and Console Output

⌵

3

Conditionals and Control Flow

⌵

4

Functions

⌵

5

Lists & Dictionaries

⌵

6

Student Becomes the Teacher

⌵

7

Lists and Functions

⌵

https://www.codecademy.com/pro?utm\_campaign=dashboard\_release&utm\_sourc...

Python | Codecademy

https://www.codecademy.com/learn/python

codecademy

Upgrade to Pro

Learn

Catalog

Dashboard

Courses

Python33%

Overview

Syllabus

1Python Syntax

2Strings and Console Output

✓Strings & Console OutputInteractive Lesson100%

✓Date and TimeInteractive Lesson100%

PROStrings and Console OutputMultiple Choice Quiz

PROPython Mad LibsFreeform Project

External Resources

Python Strings20 mins read

Basic String Operations5 mins read

# Project Ideas

- Final Project
  - Tic Tac Toe
  - Maze
  - Picture manipulation
  - FINALLY: Shape/color manipulation

sketch\_161205b ▼

```
def setup():
    size(640, 360, P3D)
    noStroke()

def draw():
    background(255, 255, 255) #sets background to white
    if keyPressed == True: #checks if key is pressed
        defineLights() #calls defineLights method
        for x in range(0, 2000, 60): #explains the range of the x axis
            for y in range(0, 2000, 60): #explains range of y axis
                with pushMatrix(): #sets up data in a matrix for (without this the shapes are in a random diagonal tube form
                    translate(x, y) #centers the shapes
                    rotateY(map(mouseX, 0, width, 0, PI))
                    rotateX(map(mouseY, 0, height, 0, PI)) #both of these define the x and y movement for the specific shapes
                    rect(25, 25, 25, 25) #creates the rectangle
            elif mousePressed: #runs section if the mouse is pressed
                defineLights()
                for x in range(0, 2000, 60):
                    for y in range(0, 2000, 60):
                        with pushMatrix():
                            translate(x, y)
                            rotateY(map(mouseX, 0, width, 0, PI))
                            rotateX(map(mouseY, 0, height, 0, PI))
                            ellipse(50, 20, 30, 23) #creates an ellipse (circle)
            elif mousePressed == False and keyPressed == False: #if neither are true
                defineLights()
                for x in range(0, 2000, 50):
                    for y in range(0, 2000, 50):
                        with pushMatrix():
```

Done saving.

Console

# Combination

A lot of what I was doing was a learning more parts of processing through Python



Python ▾

sketch\_161205b ▾

```
def setup():
    size(640, 360, P3D)
    noStroke()

def draw():
    background(255, 255, 255) #sets background to white
    if keyPressed == True: #checks if key is pressed
        defineLights() #calls defineLights method
        for x in range(0, 2000, 60): #explains the range of the x axis
            for y in range(0, 2000, 60): #explains range of y axis
                with pushMatrix(): #sets up data in a matrix for (without this the shapes are in a random diagonal tube form
                    translate(x, y) #centers the shapes
                    rotateY(map(mouseX, 0, width, 0, PI))
                    rotateX(map(mouseY, 0, height, 0, PI)) #both of these define the x and y movement for the specific shapes
                    rect(25, 25, 25, 25) #creates the rectangle
            elif mousePressed: #runs section if the mouse is pressed
                defineLights()
                for x in range(0, 2000, 60):
                    for y in range(0, 2000, 60):
                        with pushMatrix():
                            translate(x, y)
                            rotateY(map(mouseX, 0, width, 0, PI))
                            rotateX(map(mouseY, 0, height, 0, PI))
                            ellipse(50, 20, 30, 23) #creates an ellipse (circle)
            elif mousePressed == False and keyPressed == False: #if neither are true
                defineLights()
                for x in range(0, 2000, 50):
                    for y in range(0, 2000, 50):
                        with pushMatrix():
```

Console



sketch\_161205b ▼

```
def defineLights():
    for x in range(0, 2000, 60):
        for y in range(0, 2000, 60):
            with pushMatrix():
                translate(x, y)
                rotateY(map(mouseX, 0, width, 0, PI))
                rotateX(map(mouseY, 0, height, 0, PI))
                ellipse(50, 20, 30, 23) #creates an ellipse [circle]
elif mousePressed == False and keyPressed == False: #if neither are true
    defineLights()
    for x in range(0, 2000, 50):
        for y in range(0, 2000, 50):
            with pushMatrix():
                translate(x, y)
                rotateY(map(mouseX, 0, width, 0, PI))
                rotateX(map(mouseY, 0, height, 0, PI))
                triangle(50, 20, 30, 23, 34, 34) #creates a triangle

def defineLights():
    pointLight(10, 1000, 1000, 10, 1000, 1000) #green, purple, yellow, teal, blue, dark blue
    directionalLight(0, 102, 255, 1, 0, 0) #creates the different shading effects
    spotLight(255, 255, 109, 0, 40, 200, 0, -0.5, 0.5, PI / 2, 2) #more with the shading, etc.
    #deals with the different lighting, this would have been the trickiest part

#most interesting things i have learned are the different syntaxes from JAVA, etc. I was reviewing the codeSmells handout and it talked about neatness of code, (indenting, spaces, etc.)
#in python the code must be neat in order for it to run! one extra line or space will crash the code!!!
```