**Python Cheatsheet**

**Variables and Strings:**

**Hello world**

print("Hello world!")

**Hello world with a variable**

msg = "Hello world!"

print(msg)

**Concatenation (combining strings)**

first\_name = 'albert'

last\_name = 'einstein'

full\_name = first\_name + ' ' + last\_name

print(full\_name)

**Lists:**

A list stores a series of items in a particular order. You access items using an index, or within a loop.

**Make a list**

bikes = ['trek', 'redline', 'giant']

**Get the first item in a list**

first\_bike = bikes[0]

**Get the last item in a list**

last\_bike = bikes[-1]

**Looping through a list**

for bike in bikes:

print(bike)

**Adding items to a list**

bikes = []

bikes.append('trek')

bikes.append('redline')

bikes.append('giant')

**Making numerical lists**

squares = []

for x in range(1, 11):

squares.append(x\*\*2)

**Find the length of a list**

num\_users = len(users)

print("We have " + str(num\_users) + " users.")

Note:Once you've defined a list, you can change individual elements in the list. You do this by referring to the index of the item you want to modify.

**Changing an element**

users[0] = 'valerie'

users[-2] = 'ronald'

**List Contd:**

**List comprehensions**

squares = [x\*\*2 for x in range(1, 11)]

**Slicing a list**

finishers = ['sam', 'bob', 'ada', 'bea']

first\_two = finishers[:2]

**Copying a list**

copy\_of\_bikes = bikes[:]

Note: You can use the **range()** function to work with a set of numbers efficiently. The range() function starts at 0 by default, and stops one number below the number passed to it. You can use the list() function to efficiently generate a large list of numbers.

**Printing the numbers 0 to 1000**

for number in range(1001):

print(number)

**Printing the numbers 1 to 1000**

for number in range(1, 1001):

print(number)

**Making a list of numbers from 1 to a million**

numbers = list(range(1, 1000001))

Note: You can work with any set of elements from a list. A portion of a list is called a slice. To slice a list start with the index of the first item you want, then add a colon and the index after the last item you want. Leave off the first index to start at the beginning of the list, and leave off the last index to slice through the end of the list.

**Getting the first three items**

finishers = ['kai', 'abe', 'ada', 'gus', 'zoe']

first\_three = finishers[:3]

**Getting the middle three items**

middle\_three = finishers[1:4]

**Getting the last three items**

last\_three = finishers[-3:]

**Tuples:**

Tuples are similar to lists, but the items in a tuple can't be modified.

**Making a tuple**

dimensions = (1920, 1080)

**If-Else Statements:**

If statements are used to test for particular conditions and respond appropriately.

**Conditional tests**

**equals** x == 42

**not equal** x != 42

**greater than** x > 42

**or equal to** x >= 42

**less than** x < 42

**or equal to** x <= 42

**Conditional test with lists**

'trek' in bikes

'surly' not in bikes

**Assigning boolean values**

game\_active = True

can\_edit = False

**A simple if test**

if age >= 18:

print("You can vote!")

**If-elif-else statements**

if age < 4:

ticket\_price = 0

elif age < 18:

ticket\_price = 10

else:

ticket\_price = 15

**Dictionaries:**

Dictionaries store connections between pieces of information. Each item in a dictionary is a key-value pair.

**A simple dictionary**

alien = {'color': 'green', 'points': 5}

**Accessing a value**

print("The alien's color is " + alien['color'])

**Adding a new key-value pair**

alien['x\_position'] = 0

**Looping through all key-value pairs**

fav\_numbers = {'eric': 17, 'ever': 4}

for name, number in fav\_numbers.items():

print(name + ' loves ' + str(number))

**Looping through all keys**

fav\_numbers = {'eric': 17, 'ever': 4}

for name in fav\_numbers.keys():

print(name + ' loves a number')

**Looping through all the values**

fav\_numbers = {'eric': 17, 'ever': 4}

for number in fav\_numbers.values():

print(str(number) + ' is a favorite')

**User Input:**

Your programs can prompt the user for input. All input is stored as a string.

**Prompting for a value**

name = input("What's your name? ")

print("Hello, " + name + "!")

**Prompting for numerical input**

age = input("How old are you? ")

age = int(age)

pi = input("What's the value of pi? ")

pi = float(pi)

**While Loops:**

A while loop repeats a block of code as long as a condition is True.

**Counting to 5**

current\_number = 1

while current\_number <= 5:

print(current\_number)

current\_number += 1

**Letting the user choose when to quit**

prompt = "\nTell me something, and I'll "

prompt += "repeat it back to you."

prompt += "\nEnter 'quit' to end the program. "

message = ""

while message != 'quit':

message = input(prompt)

if message != 'quit':

print(message)

**Using a flag**

prompt = "\nTell me something, and I'll "

prompt += "repeat it back to you."

prompt += "\nEnter 'quit' to end the program. "

active = True

while active:

message = input(prompt)

if message == 'quit':

active = False

else:

print(message)