Python Cheat Sheet

ConnectEd Code

Input & Output

Getting, displaying and storing data.

Output a message the user can read

```
print("Hello, world!")
```

Store data in a variable and print it:

```
food = "pizza"
print("The best food is", food)
```

Ask the user for input and store the answer

```
name = input("Enter your name: ")
print("Hi", name)
```

Ask for a number and convert it to an integer

```
age = int(input("Enter age: "))
print("Next year you will turn", age + 1)
```

Use an f-string to print variables in text easily

```
name = "Leah"
year = 9
print(f"I'm {name} & I'm in year {year}!")
```

If Statements

Making decisions with code.

Check a word matches, print a message if it does

```
name = input("What's your name? ")
if name == "Alex":
  print("Alex is a great name!")
```

Choose between two options based on a number

```
number = int(input("pick a number: "))
if number == 7:
   print("I hear 7 is a lucky number!")
else:
   print("That's a normal number.")
```

Choose between multiple options

```
number = int(input("Guess a number: "))
if number < 37:
  print("Too low!")
elif number > 37:
  print("Too high!")
else:
   print("That's my favourite number!")
```

* You can use multiple **elifs** & can use if with elif without an else.

Useful Functions

Essential built-in Python functions.

Convert string to int

```
number = int("5")
```

Convert int to string

```
character = str(5)
```

Get length of a item

```
length = len("Hey")
```

Power of number (eg: 2^3)

```
cube_of_2 = pow(2, 3)
```

For Loops

Repeating code a set number of times.

Repeat loop 5 times (print numbers 0, 1, 2, 3 , 4)

```
for num in range(5):
   print(num)
```

* Remember that range starts at 0 by default.

Repeat for a specific range of numbers (eg. 3, 4, 5)

```
for num in range(3, 6):
   print(num)
```

* Remember that range doesn't include the last number.

Loop through each character of string

```
for letter in "Hello world":
   print(letter)
```

Loop through each character of string

```
for month in ["March", "April", "May"]:
   print(month)
```

While Loops

Repeating code based on a condition.

Loop forever

```
while True:
   print("Hello world!")
```

Loop until a numerical condition is met

```
num = 0
while num < 3:
   print(num)
   num = num + 1
```

Loop until the user inputs the correct answer

```
colour = input("Guess my favourite colour: ")
while colour != "Green":
   print("Wrong!")
   colour = input("Guess again: ")
print("You got it!")
```

More Conditions

To use in if statements and while loops.

Is equal to

```
age == 12
```

Is not equal

```
age != 12
```

Less than

```
age < 12
```

Less than or equal

```
age <= 12
```

Greater than

```
age > 12
```

Greater than or equal

```
age >= 12
```

Contains a string

```
"i" in "team"
```

Does not contain string

```
"i" not in "team"
```

ConnectEd Code

Connecting Coding with Context, Community & Curriculum

Lists

Creating and adding, updating & removing data.

Create an empty list

```
fruits = []
```

Create a new list with data in it

```
fruits = ["apple", "banana", "orange"]
```

Add an item to the end of the list

```
fruits.append("kiwi fruit")
```

Update an item in the list at a given index

```
fruits[0] = "pineapple"
```

Remove an item at an index

```
fruits.pop(1)
```

Remove the first occurrence of an item in a list

```
fruits.remove("orange")
```

Using lists and their values.

Access an item at a position in a list

```
item = fruits[0]
```

* Remember that in python we start counting from 0

Access an item counting from the end of the list

```
item = fruits[-1]
```

Get the index of an item in a list

```
index = fruits.index("banana")
```

Create a new sorted version of a list

```
sorted_fruits = sorted(fruits)
```

Sort the existing list

```
fruits.sort()
```

Dictionaries

Creating and updating a dictionary.

Create an empty dictionary

```
prices = {}
```

Create a dictionary with data in it (eg. prices of food)

```
prices = {"pie": 5, "apple": 0.5, "juice": 4}
```

* Dictionary keys need to be unique (a food is only included once)

Add a new item to the dictionary

```
prices["banana"] = 0.50
```

Update the value of an item in the dictionary

```
prices["pie"] = 5.5
```

Accessing data in a dictionary

Look up the value of an item in the dictionary

```
cost_apple = prices["apple"]
```

Look up a value, but get a default if it's not there

```
prices.get("orange", "Not available")
```

Loop through the keys of a dictionary

```
for food in prices:
    print(food)
```

Loop through the keys & values of a dictionary

```
for food, price in prices.items():
    print(food, price)
```

Functions

Creating and using functions for tidy & useful code.

Define a function (with no parameters) and use it

```
def print_smiley():
    print(":)")

print_smiley()
```

* Remember you need to define a function before you use it.

Define a function (1 parameter) that returns a value

```
def superfy(name):
    super_name = "Super" + name
    return super_name

print(superfy("Steph"))
print(superfy("Sam"))
```

* You can use your functions multiple times throughout your file.

Define a function (2 parameters) that returns a value

```
def new_food_maker(food1, food2):
    new_food = food1 + " " + food2
    return new_food

food = new_food_maker("chocolate", "coke")
print("I like", food)
```

* You can store values returned from a function in a variable.

Random

Essential built-in Python functions.

Import the random module

```
import random
```

* Import the random module before you use the following functions.

Get a random integer from a given range (eg. 1-10)

```
num = random.randint(1, 10)
```

* The parameters (i.e. 1-10) are both inclusive in randint.

Get a random decimal between 0 and 1.

```
decimal = random.random()
```

Choose a random item from a list

```
pick = random.choice(["win", "lose", "draw"])
```

Shuffle and existing list into a random order

```
names = ["Anna", "Ben", "Caitlin", "Denny"]
random.shuffle(names)
```