Python Cheat Sheet

Lucas Hyldgaard - TA

1 Variables and Types (Week 2)

`x = 5` – Assigns the integer value 5 to variable `x`.
`type(x)` – Returns the type of variable `x`.
`float(3)` – Converts integer 3 to float.
`int(3.9)` – Converts float 3.9 to integer (truncates).
`str(123)` – Converts integer 123 to string.
`bool(0)` – Converts value to boolean (0 → False).
`del x` – Deletes the variable `x`.
`None` – Represents the absence of a value.

2 Conditionals (Week 3)

`if condition:` – Executes a block if condition is True.
`elif condition:` – Adds another conditional branch.
`else:` – Executes a block if all conditions are False.
`==, !=, <, >, <=, >=` – Comparison operators.
`and, or, not` – Logical operators.
`in, not in` – Membership tests.
`is, is not` – Identity operators.

3 Loops (Week 4)

`for i in range(n):` – Loops `n` times.
`while condition:` – Loops while condition is True.
`break` – Exits the loop early.
`continue` – Skips the rest of the current iteration.
`for i in List` – Iterates the list for the values, NOT the indexes of the values.

`for i, val in enumerate(seq):` – Loops with index and value.
`for key, value in dict.items():` – Iterates through dictionary pairs.
`range(start, stop, step)` – Generates a sequence of numbers.
`zip(list1, list2)` – Iterates over two lists simultaneously.

4 Functions (Week 5)

`def func():` – Defines a function.
`return value` – Returns a value from a function.
`Function scope` – It is NOT possible to call on a variable within a function, UNLESS it has been returned.
`func1(func2())` – It is possible to call on a function, within another function.

5 Lists (Week 6)

`mylist = [1, 2, 3]` – Creates a list.
`mylist.append(4)` – Adds element to the end.
`mylist.insert(1, 'x')` – Inserts at index.
`mylist.remove('x')` – Removes first matching element.
`mylist.pop()` – Removes and returns last element.
`mylist.sort()` – Sorts the list in place.
`sorted(mylist)` – Returns a sorted copy.
`mylist.reverse()` – Reverses list order.
`mylist.count(x)` – Counts occurrences of `x`.
`len(mylist)` – Returns length of list.
`sum(mylist)` – Returns sum of numeric elements.
`max(mylist)` – Returns the largest value.
`min(mylist)` – Returns the smallest value.
`[x for x in range(5)]` – List comprehension.

6 Strings (Week 7)

`s = "Hello"` – Creates a string.
`s.lower()` – Converts to lowercase.
`s.upper()` – Converts to uppercase.

`s.title()` – Capitalizes each word.
`s.strip()` – Removes whitespace at both ends.
`s.replace('a', 'b')` – Replaces characters.
`s.split(',')` – Splits string into list.
`','.join(list)` – Joins list into string.
`s.find('e')` – Returns index of substring.
`s.startswith('H')` – Checks if string starts with 'H'.
`s.endswith('o')` – Checks if string ends with 'o'.
`len(s)` – Returns string length.
`s[::-1]` – Reverses the string.

7 Dictionaries and Tuples (Week 8)

`d = {'a': 1, 'b': 2}` – Creates a dictionary.
`d['a']` – Accesses value by key.
`d.get('a', 0)` – Returns value or default.
`d.keys()` – Returns all keys.
`d.values()` – Returns all values.
`d.items()` – Returns key-value pairs.
`d.update('c': 3)` – Updates dictionary.
`del d['a']` – Deletes key-value pair.
`('a', 'b', 'c')` – Tuple creation.
`t[0]` – Accesses tuple element.
`len(t)` – Tuple length.
`a, b = (1, 2)` – Tuple unpacking.
`dict(zip(keys, values))` – Builds dict from two lists.

8 Files (Week 9)

`with open('file.txt', 'r') as f:` – Opens file safely.
`f.read()` – Reads the entire file.
`f.readline()` – Reads one line.
`f.readlines()` – Reads all lines into list.
`f.write('text')` – Writes to file.
`f.close()` – Closes the file manually.
`open('file.txt', 'a')` – Opens file for appending.

9 Classes (Week 10)

`class MyClass:` – Defines a class.
`def __init__(self, x):` – Constructor method.
`self.x = x` – Assigns instance variable.
`def method(self):` – Defines instance method.
`obj = MyClass(5)` – Creates an object.
`obj.method()` – Calls a method.

10 Numpy and Matplotlib

10.1 NumPy

`import numpy as np` – Imports NumPy library.
`arr = np.array([1, 2, 3])` – Creates a 1D NumPy array.
`np.zeros(x)` – Creates an x long array filled with zeros.
`np.ones(x)` – Creates an x long array filled with ones.
`np.eye(3)` – Creates a 3x3 identity matrix.
`np.arange(0, 10, 2)` – Creates numbers from 0 to 8 with step 2.
`np.linspace(0, 1, 5)` – Generates 5 evenly spaced values between 0 and 1.
`arr.shape` – Returns dimensions of array.
`arr.reshape(2, 3)` – Reshapes array to 2x3.
`arr.ndim` – Returns number of dimensions.
`arr.size` – Returns total number of elements.
`arr.dtype` – Returns data type of array.
`arr.astype(float)` – Converts array to float type.
`np.add(a, b)` – Adds two arrays element-wise.
`np.subtract(a, b)` – Subtracts two arrays.
`np.multiply(a, b)` – Multiplies two arrays element-wise.
`np.divide(a, b)` – Divides arrays element-wise.
`np.dot(a, b)` – Matrix multiplication.
`np.transpose(a)` – Transposes an array.
`np.linalg.inv(a)` – Inverse of a matrix.
`np.linalg.det(a)` – Determinant of a matrix.

`arr.sum()` – Sum of all elements.
`arr.mean()` – Mean value of array.
`arr.std()` – Standard deviation of elements.

`arr.var()` – Variance of elements.
`np.median(arr)` – Median value.
`np.min(arr)` – Minimum value in array.
`np.max(arr)` – Maximum value in array.
`np.unique(arr)` – Returns sorted unique elements.
`np.sort(arr)` – Sorts the array.
`np.concatenate((a, b))` – Joins two arrays.
`np.vstack((a, b))` – Stacks arrays vertically.
`np.hstack((a, b))` – Stacks arrays horizontally.

`np.random.normal(0, 1, 100)` – 100 random numbers from normal distribution.
`np.random.seed(42)` – Sets random seed for reproducibility.

10.2 Matplotlib

`import matplotlib.pyplot as plt` – Imports Matplotlib plotting library.
`x = np.linspace(0, 10, 100)` – Creates X-axis data.
`y = np.sin(x)` – Example Y data using sine.

`plt.plot(x, y)` – Basic line plot.
`plt.scatter(x, y)` – Scatter plot.
`plt.bar(x, height)` – Bar chart.
`plt.hist(data, bins=10)` – Histogram with 10 bins.
`plt.boxplot(data)` – Draws a box plot.
`plt.errorbar(x, y, yerr=0.1)` – Adds error bars.
`plt.fill_between(x, y1, y2)` – Fills area between curves.
`plt.imshow(img, cmap='gray')` – Displays image data.

`plt.xlabel('X-axis')` – Sets X-axis label.
`plt.ylabel('Y-axis')` – Sets Y-axis label.
`plt.title('My Plot')` – Sets plot title.
`plt.legend()` – Shows legend.
`plt.grid(True)` – Adds grid lines.
`plt.xlim(0, 10)` – Sets X-axis range.
`plt.ylim(-1, 1)` – Sets Y-axis range.

`plt.subplot(2, 1, 1)` – Creates subplot (2 rows, 1 column, 1st plot).
`plt.figure(figsize=(8,6))` – Creates new figure with size.
`plt.tight_layout()` – Adjusts spacing automatically.

`plt.savefig('plot.png', dpi=300)` – Saves figure to file.
`plt.show()` – Displays the plot window.

```
   # Basic plotting starter example:
import numpy as np
import matplotlib.pyplot as plt
x = np.linspace(0, 2*np.pi, 100)
y = np.sin(x)
plt.plot(x, y, label='Sine Wave')
plt.xlabel('x values')
plt.ylabel('sin(x)')
plt.legend()
plt.show()
```

11 Modules (Week 13)

`import math` – Imports the math module.
`from math import sqrt` – Imports specific function.
`import random` – Imports random module.
`random.randint(1, 10)` – Random integer.
`import os` – Imports OS functions.
`os.getcwd()` – Returns current directory.
`os.listdir()` – Lists directory contents.
`import sys` – Imports system module.
`sys.exit()` – Exits program.
`import time` – Time handling.
`time.sleep(1)` – Pauses for 1 second.
`import datetime` – Imports datetime module.
`datetime.datetime.now()` – Current date and time.
`help(module)` – Displays documentation for module.

12 Godt at huske til eksamen

`Function og RETURN` – Der Ønskes næsten altid en funktion, hvorved man ALTID skal huske at lave et RETURN statement til slut.
`LÆS Igennem` – Hvilke værdier får du givet, og hvad ønskes retuneret?
`Tjek koden igennem for print()` – Man må gerne printe i løbet af ek-

samen for at tjekke hvad man er igang med at lave, men sørg for at SLETTE dem ALLE inden du afleverer!