UVa HSPC Python Cheatsheet

Primitive Data Types

`integer`	Unbounded, signed integer
`float`	53-bits of precision
`string`	Unicode string
`boolean`	True or False

Operations

`+`	Arithmetic addition or String/List concatenation
`-`	Arithmetic subtraction, set difference
`/`	Floating point division
`//`	Integer division
`%`	Integer division reminder (modulus)
`+=`	Add and update
`-=`	Subtract and update
`==`	Equality
`!=`	Inequality
`<`	Less than
`>`	Greater than
`<=`	Less than or equal
`>=`	Greater than or equal
`and`	Logical AND
`not`	Logical NOT
`or`	Logical OR

Variable Declaration and Assignment

```
index        =        0
```
NAME ASSIGNMENT VALUE

Note: Python is “duck typed,” so types are not specified at variable assignment time.

If Statement

```
if Boolean Expression :
        Statements
elif  Boolean Expression :
        Statements
else:
        Statements
```

While Loop

```
while Boolean Expression :
        Statements
```

For Loop

```
for x in Iterable:
        Statements
```

Strings

```
a = “UVa”
```
Creates the string a with value “Uva”.

```
b = “HSPC”
```
Creates the string b with value “HSPC”.

```
falseValue = a == b
```
a does not have the same value as b.

```
letterU = a[0]
```
The first character of a is the letter 'U'.

```
zero = a.index(“U”);
```
The letter “U” is the first character in the string a. Throws exception if not found.

```
substringSP = b[1:3]
```
The letter “X” does not appear in the string, returning -1.

```
String uvaHSPC = a + b;
```
The newly created string is “UVAHSPC”.

List (variable-sized Array)

```
array          =          []
```
NAME EMPTY ARRAY

```
array[index] = 50
fifty = array[index]
array.append(25) # appends 25 to end
length = len(array) # get length

# List of squares of 0 through 9
myList = [x**2 for x in range(10)]
```

Method Declaration

```
def          factorial       (n)
```
DECLARATION METHOD NAME ARGUMENTS

```
def factorial(n):
    # body
```

UVa HSPC Python Cheatsheet

Math

All return doubles. Angles, unless otherwise specified are in radians. Must "import math" to use.

`math.e`	The base of the natural logarithm.
`math.pi`	The ratio of the circumference of a circle to its diameter.
`math.degrees(rad)`	Returns the angle `rad` in degrees.
`math.radians(deg)`	Returns the angle `deg` in radians.
`math.sin(ang)`	Computes the sine of `ang`.
`math.cos(ang)`	Computes the cosine of ang.
`math.tan(ang)`	Computes the tangent of ang.
`math.asin(ang)`	Computes the inverse sine of `ang`.
`math.log(a, [base])`	The natural logarithm of `a` with respect to base `b`.
`math.sqrt(a)`	The square-root of `a`.
`math.abs(a)`	Returns the absolute value `a`.
`a**b`	Raises `a` to the power of `b`.
`max(a,b)`	Returns the maximum of `a` and `b`.
`min(a,b)`	Returns the minimum of `a` and `b`.

Reading from stdin

```
from sys import stdin
data = stdin.read().splitlines()
```

Now, data is a list of each line from stdin.

Output

```
print("I'm printing! " + str(dog));
```

Prints out a the string and the value of the variable dog with a new line.

Collections

Set

```
s = set()
```

Creates an empty set

```
s2 = { x**2 for x in range(10) }
```

Creates a set consisting of squares of 0 through 9

```
s.add(1)
```

Adds the number 1 to the set.

```
for i in s
```

Iterates through each integer in the set.

```
print(s)
```

Prints out each integer in the set.

Dictionary

```
d = {}
```

Creates an empty dictionary

```
d["Dog"] = Cat
```

Maps the string "Dog" (key) to "Cat" (value).

```
trueValue = "Cat" == d["Dog"]
```

Retrieves the value for the key "Dog" and checks for equality with "Cat".

```
for (k, v) in d.items():
  print("key: {}, value: {}".format(k,v))
```

Prints each key/value pair in the dictionary