Python - Cheat Sheet

変数宣言・初期化

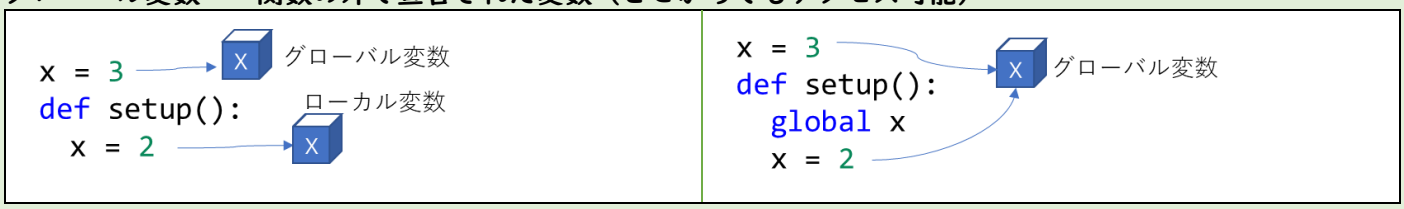
宣言	例
変数名 ＝ 初期値 1 行で複数の変数を宣言する場合は以下のように記述 変数 1，変数 2，変数 3　＝　値 1，値 2，値 3	x = 200　# x に 200 を代入 a, b = 3, 6　# a に 3、b に 6 を代入 gameOver = False　# ブール値は True か False bug = None　# 画像等は None で初期化

リスト（配列）―データを複数並べて使うときに使用

変数名 ＝ [] 変数名 ＝ [値 1，値 2，値 3，…]	images = [] heros = [None, None, None, None] scores = [1, 7, 23, 2] places = [PVector(2,3), PVector(0,1)]
変数名[インデックス]	a = scores[2]　# a は 23 scores[1] = 24 # [1,24,23,2]
追加： 変数名.append(要素) 削除： del 変数名[インデックス] 個数： len(変数名)	data = [1,3] data.append(2) # [1,3,2] del data[1]　# [1,2] len(data)　# 2

ローカル変数とグローバル変数

ローカル変数　＝関数の中で宣言された変数（関数の中のみアクセス可能）
グローバル変数　＝関数の外で宣言された変数（どこからでもアクセス可能）



※グローバル変数の値を変更する場合、関数の先頭に global 命令が必要（global がないとローカル変数扱い）

制御文―if 文と for 文

if 文（もし～なら…する）	例	コメント

if 条件～なら:
条件成立時の処理 …する
 	
※同じ値か比較は “＝＝”
（＝1 つは代入）
	if a > 3:
	println("a is bigger than 3")
	
	if c == 10:
	println("c is 10")
		if 条件 1:
		..処理 1..
		elif 条件 2:
		..処理 2..
		else:
		..処理 3..
		elif や else と組み合わせも可

for 文（繰り返す）	**例**	**コメント**

for ループ変数 in 集合:
処理
・集合の中の要素を順番に取得
・range(最小，最大，変化)も可
	for x in [2,1,4,5]:
	print(x)
	for i in range(5):
	print(i)
		ループ中で以下の処理も可能
		・break ＝ ループを抜ける
		・continue ＝ ループの先頭

for i,k in enumerate(集合): 処理 ・個々の要素を番号付で取得	data = ["apple","banana","peach"] for i, k in enumerate(data): print("i={0} k={1}".format(i, k))	出力 i=0　k=apple i=1　k=banana i=2　k=peach

Future Coders (http://www.future-coders.net) - All Rights Reserved

関数 - 複数の処理をまとめる

def　関数名（引数リスト）：
　　関数の本体
・関数の本体の範囲はインデントで明示
・値を戻すときは return 命令を使用
・return 文以降は実行されない
・引数がない場合、引数リストは空（）
	`def add(a, b):`
	`  return a + b`
	
	`def sayHello(name):`
	`  print("Hi " + name)`

リスト内包表記　－　for を使ってリストを作成する記法（およびフィルタリング）

[要素の値 for ループ変数 in 集合] ・リストを作成するので[ ]で囲む ・要素の値を記述するときにループ変数を利用可	`a = [x*2 for x in range(3)]  # [0,2,4,6]` `b = [None for i in range(10)] # None を 10 個`
[要素の値 for ループ変数 in 集合 if 条件] ・条件を満たした要素のみがリストに含まれる	`c = [i for i in range(10) if i % 2 == 0] #偶数` `d = [n for n in [1,6,2,3] if n >= 2]  #2 以上`
[要素のメソッド等 for ループ変数 in 集合] ・リストの各要素に対してメソッド呼び出しも可能	`f = [s.upper() for s in ["Hello","World"]]` `  # f は["HELLO","WORLD"]`

クラス

プロパティ:特徴	例	
class クラス名: 　　プロパティ名 ・クラス(設計図)からオブジェクト(物)を作る。 ・“クラス名()”でオブジェクトが作成される。 ・プロパティへのアクセスは 　“オブジェクト.プロパティ”と記述	宣言	`class MyCar:` `  price, fuel = 1, 2`

	利用	`a = MyCar()`
		`print(a.price)`
		`a.fuel = 200`

メソッド:クラス固有の関数	**例**	

class クラス名:
　def　メソッド名（self, 引数リスト）：
　　メソッドの本体
・メソッドの第 1 引数はオブジェクト自身
・メソッド呼び出しは
　“オブジェクト.メソッド(引数リスト)”と記述
	宣言	`class MyCar:`
		`  price, fuel = 1, 2`
		`  def drive(self, d):`
		`  self.fuel = self.fuel - d/100.0`

	利用	`a = MyCar()` `a.drive(10)`
コンストラクタ:クラスの初期化	**例**	

class クラス名 ：
　def __init__（self, 引数リスト）：
　　オブジェクトの初期化処理
 	
・コンストラクタはオブジェクト初期化に使用
・クラスの中の self は自分自身(このオブジェクト)
	宣言	`class MyCar:`
		`  price, fuel = 1, 2`
		`  def drive(self, d):`
		`  self.fuel = self.fuel - d/100.0`
		`  def __init__(self, p, f):`
		`  self.price = p`
		`  self.fuel = f`

	利用	`a = MyCar(100, 2400)` `a.drive(120)`

Future Coders (http://www.future-coders.net) - All Rights Reserved