Input and Output

Command	Structure	Example
Print	**print (** ‘ [text] ’**)**	**print(**‘Hello World’**)**
	print ([variable] **)**	**print(**myVar**)**
	print (‘ [text] [%.2f] [%s] ’ **% (** [variable,variable] **))**	**print(**‘valores: %.2f %s ’ **% (**myFloat, myStr**))**
	print (‘ **{}** [text] **{:** [.2f]**}’.format(** [variable1] , [variable2] **))**	**print (‘{}** sua idade é **{:** .2f**}’.format(**nome, idade**))**
	print (‘ [text] ’ , [variable] **)**	**print(**‘Valor: ’ , myInt**)**
	print ([variable] **+** “ [text] ” **)**	**print(**myInt **+** ‘é o valor da minha variavel’ **)**
Print in same line	**print (** [variable] **, end=**‘ ’**)**	
Input	[variable] **= input (**‘ [text] ’**)**	idade **= input(**‘Entre sua idade: ’**)**

Date and Time

Command	Structure
Import	**from datetime import datetime**
Now	[variable] = **datetime.now()**
Year	[variable]**.year**
Month	[variable]**.month**
Day	[variable]**.day**
Hour	[variable]**.hour**
MInute	[variable]**.minute**
Second	[variable]**.second**

Conditionals

Command	Structure	Obs
If	**if** [condition] **:**	
Else if	**elif** [condition] **:**	
Else	**else :**	
If-Else inline	[variable] **=** [valueTrue] **if** [condition] **else** [valueFalse]	
	print (‘ [textTrue] ’ **if** [condition] **else** [textFalse] **)**	

Loops

Command	Structure	Example	Obs
For	**for** [variable] **in range(** [number] **) :**	**for** num **in range(**myInt**):**	**num** = 0,1,2,3 para **myInt**=4
	for [variable] **in range(** [start] **,** [end] **,** [step] **):**	**for** num **in range(**4, 0, -1**):**	**num** = 4,3,2,1 (Loop decrescente)
For / Else			**else statement** é executado após o **for,** mas só se o **for** terminar normalmente (sem ter chamado um **break**)
While	**while** [condition] **:**		
While / Else	**while** [condition] **:** **else :**		**else statement** is executed after the **while,** but only if the for ends normally (not with a **break**)
Pass	**pass**		Pula bloco de comandos onde está
Continue	**continue**		Passa para a próxima iteração do loop
Break	**break**		Força a **sair do loop**

Tratamento de Erros

Command	Structure	Example	Obs

Try
Except
	try:
	  [statements]
	except:
	  [statements_if_error]
		try:
		  print (2 **+** ‘casa’ **)**
		except:
		  print (‘aconteceu um errro’ **)**
			Tenta realizar o bloco de comando
			se ocorrer um erro, o erro é capturado e é executado os comandos do ‘except’

	try:
	  [statements]
	except [errorFound] **:**
	  [statements_if_error]
	except [errorFound2] **:**
	  [statements_if_error2]
			Define um tipo de erro específico
			É **possível** ter **vários except**, cada erro tem um bloco especifico

	try:
	  [statements]
	except ([errorFound] **,** [errorFound2] **):**
	  [statements_if_error ou _if_error2]
			Vários erros chamam o mesmo bloco

	try:
	  [statements]
	except [errorFound] **as** [variableError] **:**
	  [statements_if_error]
			Atribui a uma **variável** o **erro encontrado** para usa-lo no bloco de comandos seguinte

Try
Except
Finnary
	try:
	  [statements]
	except:
	  [statements_if_error]
	finally:
	  [statementsFinally]
			Comandos em **finally** são **executados sempre** no final, independentemente se houve erro ou não

Try
Except
Else
Finnary
	try:
	  [statements]
	except:
	  [statements_if_error]
	else:
	  [statements_if_suceeded]
	finally:
	  [statementsFinally]
			Comandos em **else** são **executados** após o try, **se não houve erro**

Raise	**raise** [type_of_error]		
	raise [type_of_error] **(** ‘ [mensagem] ’ **)**	a **= input(**‘Entre num de 1 a 10’**)** **if not 1<= a <= 10:** **raise** ValueError**(** ‘mensagem’ **)**	Cria um erro
Tipos de Erros	**TypeError**		
	KeyError		
	ValueError		
	IndexError		