

APÊNDICE B – ELEMENTOS DA LINGUAGEM PYTHON

Palavras reservadas					
and	def	exec	if	not	raise
assert	del	finally	import	or	try
break	elif	for	in	pass	while
class	else	from	is	print	yield
continue	except	global	lambda	return	

Funções embutidas no core (Built-in)				
<u>abs()</u>	<u>divmod()</u>	<u>input()</u>	<u>open()</u>	<u>staticmethod()</u>
<u>all()</u>	<u>enumerate()</u>	<u>int()</u>	<u>ord()</u>	<u>str()</u>
<u>any()</u>	<u>eval()</u>	<u>isinstance()</u>	<u>pow()</u>	<u>sum()</u>
<u>basestring()</u>	<u>execfile()</u>	<u>issubclass()</u>	<u>print()</u>	<u>super()</u>
<u>bin()</u>	<u>file()</u>	<u>iter()</u>	<u>property()</u>	<u>tuple()</u>
<u>bool()</u>	<u>filter()</u>	<u>len()</u>	<u>range()</u>	<u>type()</u>
<u>bytearray()</u>	<u>float()</u>	<u>list()</u>	<u>raw_input()</u>	<u>unichr()</u>
<u>callable()</u>	<u>format()</u>	<u>locals()</u>	<u>reduce()</u>	<u>unicode()</u>
<u>chr()</u>	<u>frozenset()</u>	<u>long()</u>	<u>reload()</u>	<u>vars()</u>
<u>classmethod()</u>	<u>getattr()</u>	<u>map()</u>	<u>repr()</u>	<u>xrange()</u>
<u>cmp()</u>	<u>globals()</u>	<u>max()</u>	<u>reversed()</u>	<u>zip()</u>
<u>compile()</u>	<u>hasattr()</u>	<u>memoryview()</u>	<u>round()</u>	<u>import ()</u>
<u>complex()</u>	<u>hash()</u>	<u>min()</u>	<u>set()</u>	
<u>delattr()</u>	<u>help()</u>	<u>next()</u>	<u>setattr()</u>	
<u>dict()</u>	<u>hex()</u>	<u>object()</u>	<u>slice()</u>	
<u>dir()</u>	<u>id()</u>	<u>oct()</u>	<u>sorted()</u>	

Operadores aritméticos	Operadores comparativos	Operadores lógicos
+	< (menor que)	or
-	> (maior que)	and
/ (divisão)	<= (menor igual que)	not
* (multiplicação)	>= (maior igual que)	in
** (exponenciação)	!= (diferente)	is
% (resto da divisão)	== (igual)	is not

Variáveis	Descrição	Exemplo de Sintaxe
Inteiro	Número com precisão infinita	>>> a= 2
Float	Número com precisão declarada	>>> a= 2.4456

Complex	Número bidimensional	>>> a= 2+3.2j >>> b= complex(2, 3.2)
Boolean	Verdadeiro ou Falso	>>> a= True >>> b= False
String Unicode	Conjunto ordenado de caracteres Conjunto ordenado de caracteres	>>> a= 'Ola primeira dama' >>> b= u'Ol\u00E9 1\u00AA dama'
List	Conjunto ordenado mutável de elementos	>>> a= [1.0, 2, 'torta'] >>> b= [1, [2.1, 2.2, 2.3], 3]
Tuple	Conjunto ordenado imutável de elementos	>>> a= (2, "") >>> b= (0, (1.1, 1.2), 2)
Dicionário	Conjunto de atributos com seus valores (chave:conteúdo)	>>> a= {'nome': "Terezinha", 'nota':10} >>> b= dict(nome='Juninho', nota=0)

Laços	Descrição	Exemplo Input-Output (a=5)
if... elif... else...	Se... Ou se... Caso contrário...	>>> if a==4: >>> print "bolacha" >>> else: >>> print "biscoito" biscoito
for... in	Percorre cada elemento de uma variável iterável (list ou tuple) ou sobre as chaves de um dicionário	>>> for i in ['Zefinha', 'tem', a+10, 'filhos']: >>> print i , Zefinha tem 15 filhos >>>for i in { 'Nome': 'img', u'Extens\u00E3o': '.png' }: >>> print i , Extensão Nome
while...	Repete uma operação enquanto um valor for verdadeiro.	>>> while a>=0: >>> a=a-1 >>> print a, 4 3 2 1 0 -1
break	Termina/escapa de um loop (for ou while), interrompendo-o	>>> while a>=0: >>> a=a-1 >>> if a==2: >>> break >>> print a, 4 3
continue	Pula a porção seguinte do loop (for ou while) mas não o interrompe	>>> while a>=0: >>> a=a-1 >>> if a==2: >>> continue >>> print a, 4 3 1 0 -1
try... except...	Tenta executar um comando, caso originar algum erro executa outro comando	>>> try: >>> b=a/0 >>> except: >>> b=a/5 >>> print b 1
def...	Define uma função.	>>> def soma(x, y):

return	Argumentos de entrada ou saída são opcionais	<pre> >>> return x+y >>> print soma(a,2) 7 </pre>
class...	Define uma classe.	<pre> >>> class aluno: >>> nome='' >>> def soma(self,x,y): >>> return x+y >>> objeto_alguem=aluno() >>> objeto_alguem.nome='Huguinho' >>> print objeto_alguem.nome, >>> print "somou 'a' com 2 e respondeu", >>> print objeto_alguem.soma(a,2) Huguinho somou 'a' com 2 e respondeu 7 </pre>