

Laboratório de Programação com Games

Professor:

Anselmo Montenegro
www.ic.uff.br/~anselmo

Conteúdo:

- Conceitos de programação com Python

Roteiro

- Introdução
- O conceito de algoritmos
- A Linguagem Python
- Representando e manipulando dados: literais, variáveis e expressões
- Entrada e saída de dados
- Tomando decisões: estruturas de seleção
- Repetindo comandos: estrutura de repetição
- Agregando valores: listas/arrays
- Um primeiro nível de modularização: funções
- Um exemplo de jogo com interface textual

Introdução

- Um computador é um dispositivo eletrônico capaz de resolver problemas e realizar diferentes tarefas sequencialmente ou concomitantemente
- Atualmente os computadores são ubíquos, estando presente em quase todas as tarefas e atividades que realizamos
- Encontramos o uso dos computadores não somente em sistemas de informação, mas também como parte integrante de automóveis, celulares, TVs, etc.

Introdução

- Para realizar qualquer tarefa ou resolver um problema, um computador recebe como entrada um conjunto de dados sobre os quais são aplicados um conjunto de operações bem definidas
- Os dados são parte da descrição do problema que se deseja resolver ou tarefa que se deseja realizar

Introdução

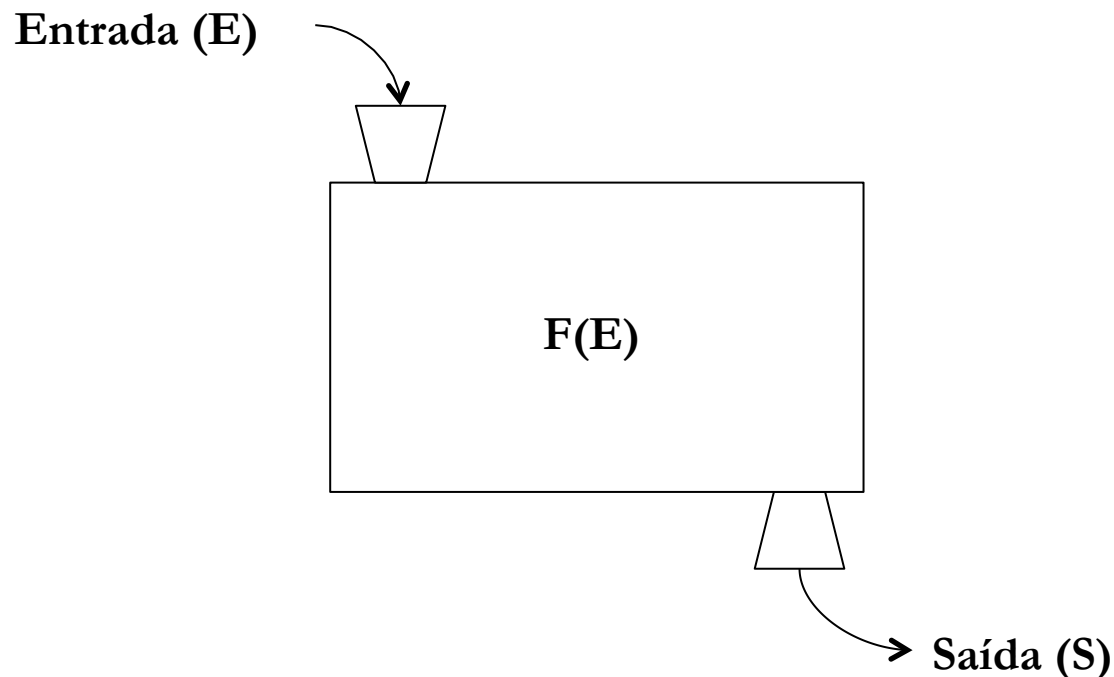
- O processamento dos dados ocorre quando são aplicadas as operações sobre os dados com fins de obter algum resultado
- Tarefas complexas nada mais são que uma combinação apropriada de uma sequencia bem definida de operações, aplicadas aos dados de entrada

Noção de algoritmo

- Uma das funções do computador é essencialmente resolver problemas
- Um problema é descrito por um enunciado que relaciona um conjunto de dados de entrada com um conjunto de dados de saída, a resposta do problema
- Exemplo: Dado uma sequência de inteiros de 1 a 10, onde os números encontram-se em posição arbitrária, colocá-los em ordem crescente

Noção de algoritmo

- Resolver um problema consiste em encontrar uma relação entre os dados de entrada e os dados de saída que satisfaça o enunciado
- Esta relação pode ser modelada como uma função

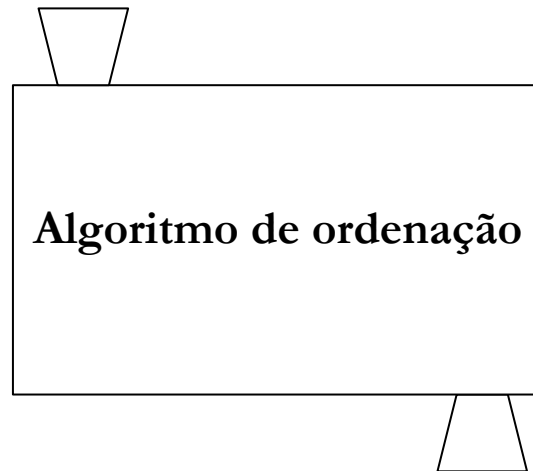


Noção de algoritmo

- Na maioria dos casos não é possível encontrar uma função matemática, no sentido convencional, que mapeie cada dado de entrada nos seu dado correto no conjunto de saída
- Entretanto a solução pode ser obtida através de um *algoritmo*
- Um *algoritmo* é uma sequência de passos bem definidos e finitos que levam a solução de um problema

Noção de algoritmo

9,6,8,7,5, 3, 4, 1, 0, 2



0, 1, 2, 3, 4, 5,6,7,8,9

Noção de algoritmo

- Um exemplo de algoritmo são as instruções para trocar o pneu de um carro:

1. Ligue o pisca-alerta e estacione
2. Engate a primeira marcha
3. Puxe o freio-de-mão
4. Saia do carro
5. Retire o triangulo do porta-malas
6. Numero de passos = velocidade máxima da via
7. Se houver chuva ou neblina:
 8. numero de passos = numero de passos x 2
9. Coloque o triângulo após andar numero de passos
10. Pegue o macaco, estepe e chave de roda e posicione-os próximo ao pneu furado
11. Posicione o macaco embaixo do carro
12. Se o macaco for de manivela então:
 13. Enquanto a altura correta não for atingida:
 14. gire a manivela
 15. Senão:
 16. Enquanto a altura correta não for atingida:
 17. aplique pressão

18. Para cada uma das quatro porcas:
 19. Coloque a chave-de-roda na porca
 20. Enquanto a porca não se soltar:
 21. Gire a chave-de-roda
 22. Se o macaco for de manivela então:
 23. Enquanto a roda não sair do chão:
 24. gire a manivela
 25. Senão:
 26. Enquanto a roda não sair do chão:
 27. aplique pressão
 28. Retire o pneu
 29. Coloque estepe
 30. Para cada uma das quatro porcas:
 31. Coloque a chave-de-roda na porca
 32. Enquanto não sentir a porca apertada:
 33. Gire a chave-de-roda
 34. Desça o macaco com cuidado
 35. Guarde os equipamentos de troca
 36. Entre e saia com o carro

Noção de algoritmo

- Apesar do algoritmo anterior podemos identificar praticamente todas as construções que usaremos para definir algoritmos computacionais

1. Ligue o pisca-alerta e estacione
2. Engate a primeira marcha
3. Puxe o freio-de-mão
4. Saia do carro
5. Retire o triangulo do porta-malas
6. Numero de passos = velocidade máxima da via
7. Se houver chuva ou neblina:
 8. numero de passos = numero de passos x 2
9. Coloque o triângulo após andar numero de passos
10. Pegue o macaco, estepe e chave de roda e posicione-os próximo ao pneu furado
11. Posicione o macaco embaixo do carro
12. Se o macaco for de manivela então:
 13. Enquanto a altura correta não for atingida:
 14. aplique pressão
 15. Senão:
 16. Enquanto a altura correta não for atingida:
 17. gire a manivela

18. Para cada uma das quatro porcas:
 19. Coloque a chave-de-roda na porca
 20. Enquanto a porca não se soltar:
 21. Gire a chave-de-roda
 22. Se o macaco for de manivela então:
 23. Enquanto a roda não sair do chão:
 24. aplique pressão
 25. Senão:
 26. Enquanto a roda não sair do chão:
 27. gire a manivela
 28. Retire o pneu
 29. Coloque estepe
 30. Para cada uma das quatro porcas:
 31. Coloque a chave-de-roda na porca
 32. Enquanto não sentir a porca apertada:
 33. Gire a chave-de-roda
 34. Desça o macaco com cuidado
 35. Guarde os equipamentos de troca
 36. Entre e saia com o carro

Noção de algoritmo

- Fluxo seqüencial: as instruções ocorrem naturalmente uma após as outras se nada for indicado

1. Ligue o pisca-alerta e estacione
2. Engate a primeira marcha
3. Puxe o freio-de-mão
4. Saia do carro
5. Retire o triangulo do porta-malas
6. Numero de passos = velocidade máxima da via
7. Se houver chuva ou neblina:
 8. numero de passos = numero de passos x 2
9. Coloque o triângulo após andar numero de passos
10. Pegue o macaco, estepe e chave de roda e posicione-os próximo ao pneu furado
11. Posicione o macaco embaixo do carro
12. Se o macaco for de manivela então:
 13. Enquanto a altura correta não for atingida:
 14. aplique pressão
 15. Senão:
 16. Enquanto a altura correta não for atingida:
 17. gire a manivela

18. Para cada uma das quatro porcas:
 19. Coloque a chave-de-roda na porca
 20. Enquanto a porca não se soltar:
 21. Gire a chave-de-roda
 22. Se o macaco for de manivela então:
 23. Enquanto a roda não sair do chão:
 24. aplique pressão
 25. Senão:
 26. Enquanto a roda não sair do chão:
 27. gire a manivela
 28. Retire o pneu
 29. Coloque estepe
 30. Para cada uma das quatro porcas:
 31. Coloque a chave-de-roda na porca
 32. Enquanto não sentir a porca apertada:
 33. Gire a chave-de-roda
 34. Desça o macaco com cuidado
 35. Guarde os equipamentos de troca
 36. Entre e saia com o carro

Noção de algoritmo

- Busca de objetos para que uma operação possa ser realizada

1. Ligue o pisca-alerta e estacione
2. Engate a primeira marcha
3. Puxe o freio-de-mão
4. Saia do carro
5. Retire o triangulo do porta-malas
6. Numero de passos = velocidade máxima da via
7. Se houver chuva ou neblina:
 8. numero de passos = numero de passos x 2
9. Coloque o triângulo após andar numero de passos
10. Pegue o macaco, estepe e chave de roda e posicione-os próximo ao pneu furado
11. Posicione o macaco embaixo do carro
12. Se o macaco for de manivela então:
 13. Enquanto a altura correta não for atingida:
 14. aplique pressão
 15. Senão:
 16. Enquanto a altura correta não for atingida:
 17. gire a manivela

18. Para cada uma das quatro porcas:
 19. Coloque a chave-de-roda na porca
 20. Enquanto a porca não se soltar:
 21. Gire a chave-de-roda
 22. Se o macaco for de manivela então:
 23. Enquanto a roda não sair do chão:
 24. aplique pressão
 25. Senão:
 26. Enquanto a roda não sair do chão:
 27. gire a manivela
 28. Retire o pneu
 29. Coloque estepe
 30. Para cada uma das quatro porcas:
 31. Coloque a chave-de-roda na porca
 32. Enquanto não sentir a porca apertada:
 33. Gire a chave-de-roda
 34. Desça o macaco com cuidado
 35. Guarde os equipamentos de troca
 36. Entre e saia com o carro

Noção de algoritmo

- Atribuição de valores a uma variável

1. Ligue o pisca-alerta e estacione
2. Engate a primeira marcha
3. Puxe o freio-de-mão
4. Saia do carro
5. Retire o triangulo do porta-malas
6. **Numero de passos = velocidade máxima da via**
7. Se houver chuva ou neblina:
 8. numero de passos = numero de passos x 2
9. Coloque o triângulo após andar numero de passos
10. Pegue o macaco, estepe e chave de roda e posicione-os próximo ao pneu furado
11. Posicione o macaco embaixo do carro
12. Se o macaco for de manivela então:
 13. Enquanto a altura correta não for atingida:
 14. aplique pressão
 15. Senão:
 16. Enquanto a altura correta não for atingida:
 17. gire a manivela

18. Para cada uma das quatro porcas:
 19. Coloque a chave-de-roda na porca
 20. Enquanto a porca não se soltar:
 21. Gire a chave-de-roda
 22. Se o macaco for de manivela então:
 23. Enquanto a roda não sair do chão:
 24. aplique pressão
 25. Senão:
 26. Enquanto a roda não sair do chão:
 27. gire a manivela
 28. Retire o pneu
 29. Coloque estepe
 30. Para cada uma das quatro porcas:
 31. Coloque a chave-de-roda na porca
 32. Enquanto não sentir a porca apertada:
 33. Gire a chave-de-roda
 34. Desça o macaco com cuidado
 35. Guarde os equipamentos de troca
 36. Entre e saia com o carro

Noção de algoritmo

- Avaliação de expressões

1. Ligue o pisca-alerta e estacione
2. Engate a primeira marcha
3. Puxe o freio-de-mão
4. Saia do carro
5. Retire o triangulo do porta-malas
6. Numero de passos = velocidade máxima da via
7. Se houver chuva ou neblina:
8. numero de passos = **numero de passos x 2**
9. Coloque o triângulo após andar numero de passos
10. Pegue o macaco, estepe e chave de roda e posicione-os próximo ao pneu furado
11. Posicione o macaco embaixo do carro
12. Se o macaco for de manivela então:
13. Enquanto a altura correta não for atingida:
14. aplique pressão
15. Senão:
16. Enquanto a altura correta não for atingida:
17. gire a manivela

18. Para cada uma das quatro porcas:
19. Coloque a chave-de-roda na porca
20. Enquanto a porca não se soltar:
21. Gire a chave-de-roda
22. Se o macaco for de manivela então:
23. Enquanto a roda não sair do chão:
24. aplique pressão
25. Senão:
26. Enquanto a roda não sair do chão:
27. gire a manivela
28. Retire o pneu
29. Coloque estepe
30. Para cada uma das quatro porcas:
31. Coloque a chave-de-roda na porca
32. Enquanto não sentir a porca apertada:
33. Gire a chave-de-roda
34. Desça o macaco com cuidado
35. Guarde os equipamentos de troca
36. Entre e saia com o carro

Noção de algoritmo

- Desvio de fluxo de instruções: o fluxo das instruções são alterados em função de alguma condição

1. Ligue o pisca-alerta e estacione
2. Engate a primeira marcha
3. Puxe o freio-de-mão
4. Saia do carro
5. Retire o triangulo do porta-malas
6. Numero de passos = velocidade máxima da via
7. **Se houver chuva ou neblina:**
8. **numero de passos = numero de passos x 2**
9. Coloque o triângulo após andar numero de passos
10. Pegue o macaco, estepe e chave de roda e posicione-os próximo ao pneu furado
11. Posicione o macaco embaixo do carro
12. Se o macaco for de manivela então:
13. Enquanto a altura correta não for atingida:
14. aplique pressão
15. Senão:
16. Enquanto a altura correta não for atingida:
17. gire a manivela

18. Para cada uma das quatro porcas:
19. Coloque a chave-de-roda na porca
20. Enquanto a porca não se soltar:
21. Gire a chave-de-roda
22. Se o macaco for de manivela então:
23. Enquanto a roda não sair do chão:
24. aplique pressão
25. Senão:
26. Enquanto a roda não sair do chão:
27. gire a manivela
28. Retire o pneu
29. Coloque estepe
30. Para cada uma das quatro porcas:
31. Coloque a chave-de-roda na porca
32. Enquanto não sentir a porca apertada:
33. Gire a chave-de-roda
34. Desça o macaco com cuidado
35. Guarde os equipamentos de troca
36. Entre e saia com o carro

Noção de algoritmo

- Desvio de fluxo de instruções 2: o fluxo das instruções são alterados em função de alguma condição

1. Ligue o pisca-alerta e estacione
2. Engate a primeira marcha
3. Puxe o freio-de-mão
4. Saia do carro
5. Retire o triangulo do porta-malas
6. Numero de passos = velocidade máxima da via
7. Se houver chuva ou neblina:
 8. numero de passos = numero de passos x 2
9. Coloque o triângulo após andar numero de passos
10. Pegue o macaco, estepe e chave de roda e posicione-os próximo ao pneu furado
11. Posicione o macaco embaixo do carro
12. **Se o macaco for de manivela então:**
13. **Enquanto a altura correta não for atingida:**
14. **aplique pressão**
15. **Senão:**
16. **Enquanto a altura correta não for atingida:**
17. **gire a manivela**

18. Para cada uma das quatro porcas:
19. Coloque a chave-de-roda na porca
20. Enquanto a porca não se soltar:
21. Gire a chave-de-roda
22. **Se o macaco for de manivela então:**
23. **Enquanto a roda não sair do chão:**
24. **aplique pressão**
25. **Senão:**
26. **Enquanto a roda não sair do chão:**
27. **gire a manivela**
28. Retire o pneu
29. Coloque estepe
30. Para cada uma das quatro porcas:
31. Coloque a chave-de-roda na porca
32. Enquanto não sentir a porca apertada:
33. Gire a chave-de-roda
34. Desça o macaco com cuidado
35. Guarde os equipamentos de troca
36. Entre e saia com o carro

Noção de algoritmo

- Repetição com número de vezes conhecido

1. Ligue o pisca-alerta e estacione
2. Engate a primeira marcha
3. Puxe o freio-de-mão
4. Saia do carro
5. Retire o triangulo do porta-malas
6. Numero de passos = velocidade máxima da via
7. Se houver chuva ou neblina:
 8. numero de passos = numero de passos x 2
9. Coloque o triângulo após andar numero de passos
10. Pegue o macaco, estepe e chave de roda e posicione-os próximo ao pneu furado
11. Posicione o macaco embaixo do carro
12. Se o macaco for de manivela então:
 13. Enquanto a altura correta não for atingida:
 14. aplique pressão
 15. Senão:
 16. Enquanto a altura correta não for atingida:
 17. gire a manivela

18. Para cada uma das quatro porcas:
 19. Coloque a chave-de-roda na porca
 20. Enquanto a porca não se soltar:
 21. Gire a chave-de-roda
 22. Se o macaco for de manivela então:
 23. Enquanto a roda não sair do chão:
 24. aplique pressão
 25. Senão:
 26. Enquanto a roda não sair do chão:
 27. gire a manivela
 28. Retire o pneu
 29. Coloque estepe
 30. Para cada uma das quatro porcas:
 31. Coloque a chave-de-roda na porca
 32. Enquanto não sentir a porca apertada:
 33. Gire a chave-de-roda
 34. Desça o macaco com cuidado
 35. Guarde os equipamentos de troca
 36. Entre e saia com o carro

Noção de algoritmo

- Repetição condicional

1. Ligue o pisca-alerta e estacione
2. Engate a primeira marcha
3. Puxe o freio-de-mão
4. Saia do carro
5. Retire o triangulo do porta-malas
6. Numero de passos = velocidade máxima da via
7. Se houver chuva ou neblina:
 8. numero de passos = numero de passos x 2
9. Coloque o triângulo após andar numero de passos
10. Pegue o macaco, estepe e chave de roda e posicione-os próximo ao pneu furado
11. Posicione o macaco embaixo do carro
12. Se o macaco for de manivela então:
 13. Enquanto a altura correta não for atingida:
 14. aplique pressão
 15. Senão:
 16. Enquanto a altura correta não for atingida:
 17. gire a manivela

18. Para cada uma das quatro porcas:
 19. Coloque a chave-de-roda na porca
 20. Enquanto a porca não se soltar:
 21. Gire a chave-de-roda
 22. Se o macaco for de manivela então:
 23. Enquanto a roda não sair do chão:
 24. aplique pressão
 25. Senão:
 26. Enquanto a roda não sair do chão:
 27. gire a manivela
 28. Retire o pneu
 29. Coloque estepe
 30. Para cada uma das quatro porcas:
 31. Coloque a chave-de-roda na porca
 32. Enquanto não sentir a porca apertada:
 33. Gire a chave-de-roda
 34. Desça o macaco com cuidado
 35. Guarde os equipamentos de troca
 36. Entre e saia com o carro

Noção de algoritmo

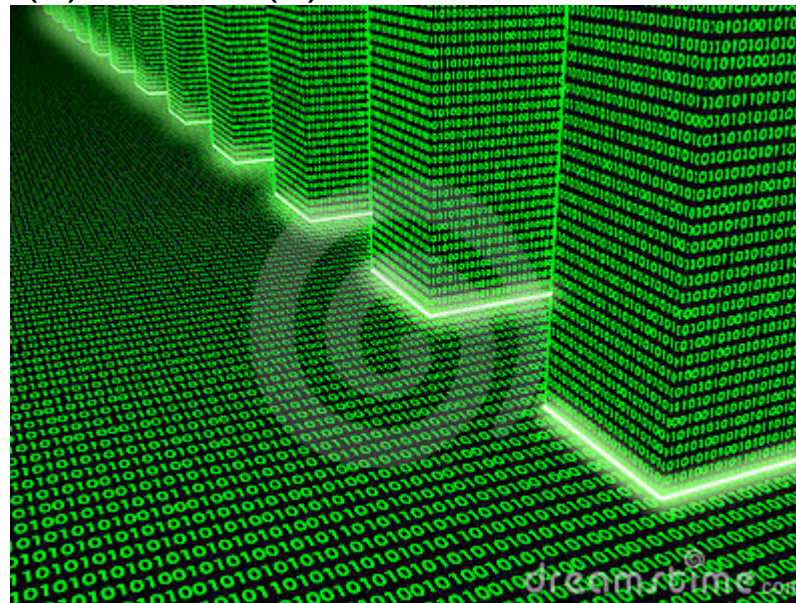
- Resumindo, iremos estudar os seguintes componentes de um algoritmo e como expressá-lo em uma linguagem de programação:
- Uso de variáveis e constantes
- Atribuição de valores
- Comandos de desvio de fluxo
- Repetições
- Definição e uso de funções
- Representação e manipulação de agregados de dados

Introdução à programação

- Programar um computador significa alimentá-lo com dados sobre os quais são executados algoritmos
- Os algoritmos devem ser definidos através de operações que o computador seja capaz de executar

Introdução à programação

- Computadores são máquinas digitais que codificam dados e instruções através de códigos binários
- Os valores binários correspondem fisicamente a limiares de voltagem em dois níveis, alto (0) e baixo (1)



<http://www.dreamstime.com/stock-photo-binary-matrix-01-image552370>

Introdução à programação

- Nos primórdios da computação, os algoritmos era codificados diretamente nas instruções da máquina
- Isso era extremamente tedioso e sujeito a erros
- Com o tempo, foram desenvolvidas linguagens de programação mais próximas as linguagens usadas pelos humanos: linguagens de alto nível
- Para utilizar as linguagens de alto nível é necessário um processo de tradução entre o código em alto nível e a linguagem da máquina

Introdução à programação

- Existem dois processos de tradução: interpretação e compilação
- Na compilação todo o código na linguagem de alto nível é analisado de uma só vez e convertido em instruções da máquina alvo, que posteriormente é executado
- Na interpretação, as instruções são analisadas uma-a-uma e executadas

Introdução à programação

- Python é uma linguagem de alto nível interpretada com as seguintes características:



- Simples
- Fácil aprendizado
- Versátil

Introdução à programação

- Representação e manipulação de dados:
 - Para programar precisamos de dados.
 - Os dados podem ser de diferentes naturezas, podendo ser primitivos ou compostos

Introdução à programação

- Todo dado tem um tipo, isto é, ele pertence a um conjunto para o qual são definidas operações válidas
- Começemos com os tipos mais básicos da linguagem Python:
 - Inteiro (int): 0, -893, 3, -4, 1739, ...
 - Real (float): 0.64, -587.0, 3.1415..., -85423677.5468782, ...
 - Caractere (char): "a", " ", "4", "&", "G", ".", ",", ...
 - Booleano (boolean): true, false

Introdução à programação

- Ao realizar operações, precisamos analisar sua coerência
- Por exemplo, não faz sentido somarmos um valor booleano a um número inteiro
- Tão pouco faz sentido multiplicar um caractere “A” ao número 1.34
- O que significaria tais operações? Elas são inválidas

Introdução à programação

- Que tipo de operações podemos realizar com cada um dos tipos?
 - Operadores aritméticos
 - Operadores relacionais
 - Operadores de atribuição
 - Operadores lógicos
 - Operadores bit-a-bit (bitwise)
 - Operadores de inclusão
 - Operadores de identidade

Introdução à programação

Operadores Aritméticos		
Operator	Description	Example: a=10, b=20
+	Addition - Adds values on either side of the operator	a + b will give 30
-	Subtraction - Subtracts right hand operand from left hand operand	a - b will give -10
*	Multiplication - Multiplies values on either side of the operator	a * b will give 200
/	Division - Divides left hand operand by right hand operand	b / a will give 2
%	Modulus - Divides left hand operand by right hand operand and returns remainder	b % a will give 0
**	Exponent - Performs exponential (power) calculation on operators	a**b will give 10 to the power 20
//	Floor Division - The division of operands where the result is the quotient in which the digits after the decimal point are removed.	9//2 is equal to 4 and 9.0//2.0 is equal to 4.0

Introdução à programação

Operadores Relacionais		
Operator	Description	Example: a=10, b=20
==	Checks if the value of two operands are equal or not, if yes then condition becomes true.	(a == b) is not true.
!=	Checks if the value of two operands are equal or not, if values are not equal then condition becomes true.	(a != b) is true.
<>	Checks if the value of two operands are equal or not, if values are not equal then condition becomes true.	(a <> b) is true. This is similar to != operator.
>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	(a > b) is not true.
<	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	(a < b) is true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	(a >= b) is not true.
<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	(a <= b) is true

Introdução à programação

Operadores de Atribuição		
Operator	Description	Example: a=10, b=20
=	Simple assignment operator, Assigns values from right side operands to left side operand	$c = a + b$ will assigned value of $a + b$ into c
+=	Add AND assignment operator, It adds right operand to the left operand and assign the result to left operand	$c += a$ is equivalent to $c = c + a$
-=	Subtract AND assignment operator, It subtracts right operand from the left operand and assign the result to left operand	$c -= a$ is equivalent to $c = c - a$
*=	Multiply AND assignment operator, It multiplies right operand with the left operand and assign the result to left operand	$c *= a$ is equivalent to $c = c * a$
/=	Divide AND assignment operator, It divides left operand with the right operand and assign the result to left operand	$c /= a$ is equivalent to $c = c / a$
%=	Modulus AND assignment operator, It takes modulus using two operands and assign the result to left operand	$c \% = a$ is equivalent to $c = c \% a$
**=	Exponent AND assignment operator, Performs exponential (power) calculation on operators and assign value to the left operand	$c ** = a$ is equivalent to $c = c ** a$
//=	Floor Division and assigns a value, Performs floor division on operators and assign value to the left operand	$c //= a$ is equivalent to $c = c // a$

Introdução à programação

Operadores lógicos		
Operator	Description	Example
and	Called Logical AND operator. If both the operands are true then then condition becomes true.	(a and b) is true.
or	Called Logical OR Operator. If any of the two operands are non zero then then condition becomes true.	(a or b) is true.
not	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false.	not(a and b) is false.

Introdução à programação

Operadores Bit-a-bit		
Operator	Description	Example: a = 0011 1100 (60) b = 0000 1101 (13)
&	Binary AND Operator copies a bit to the result if it exists in both operands.	(a & b) will give 12 which is 0000 1100
	Binary OR Operator copies a bit if it exists in either operand.	(a b) will give 61 which is 0011 1101
^	Binary XOR Operator copies the bit if it is set in one operand but not both.	(a ^ b) will give 49 which is 0011 0001
~	Binary Ones Complement Operator is unary and has the effect of 'flipping' bits.	(~a) will give -61 which is 1100 0011 in 2's complement form due to a signed binary number.
<<	Binary Left Shift Operator. The left operands value is moved left by the number of bits specified by the right operand.	a << 2 will give 240 which is 1111 0000
>>	Binary Right Shift Operator. The left operands value is moved right by the number of bits specified by the right operand.	a >> 2 will give 15 which is 0000 1111

Introdução à programação

Operadores lógicos		
Operator	Description	Example
and	Called Logical AND operator. If both the operands are true then then condition becomes true.	(a and b) is true.
or	Called Logical OR Operator. If any of the two operands are non zero then then condition becomes true.	(a or b) is true.
not	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false.	not(a and b) is false.

Introdução à programação

Operador de Inclusão		
Operator	Description	Example
in	Evaluates to true if it finds a variable in the specified sequence and false otherwise.	x in y, here in results in a 1 if x is a member of sequence y.
not in	Evaluates to true if it does not finds a variable in the specified sequence and false otherwise.	x not in y, here not in results in a 1 if x is not a member of sequence y

Introdução à programação

Operador de identidade		
Operator	Description	Example
is	Evaluates to true if the variables on either side of the operator point to the same object and false otherwise.	x is y, here is results in 1 if id(x) equals id(y).
is not	Evaluates to false if the variables on either side of the operator point to the same object and true otherwise.	x is not y, here is not results in 1 if id(x) is not equal to id(y).

Introdução à programação

Precedência de operadores (maior de cima para baixo)	
Operator	Description
**	Exponentiation (raise to the power)
~ + -	Ccomplement, unary plus and minus (method names for the last two are +@ and -@)
* / % //	Multiply, divide, modulo and floor division
+ -	Addition and subtraction
>> <<	Right and left bitwise shift
&	Bitwise 'AND'
^	Bitwise exclusive 'OR' and regular 'OR'
<= < > >=	Comparison operators
<> == !=	Equality operators
= %= /= //= -= += *= **=	Assignment operators
is is not	Identity operators
in not in	Membership operators
not or and	Logical operators

Introdução à programação

- Os dados precisam ser armazenados em algum lugar para que possam ser manipulados
- Sabemos que eles são armazenados em memória

Introdução à programação

- Como podemos definir dados?
 - Definindo um valor fixo no código fonte (um literal)
`10 + 3`
 - Atribuindo um valor a uma variável ou constante
`x = 10.5`
`GRAVIDADE = 9.8`
 - Em Python, diferentemente de outras linguagens, constantes são tratadas como variáveis, apenas não muda seus valores

Introdução à programação

- Quando definimos um literal o interpretador implicitamente reserva um lugar na memória para armazená-lo
- Uma variável por sua vez é apenas um rótulo, ou um endereço da posição de memória que armazena um valor

Introdução à programação

- Como declaramos uma variável?
- Uma variável é representada por um identificador, isto é um símbolo formado por caracteres
- Um identificador é um símbolo que rotula algum objeto da linguagem: variáveis, constantes, funções, classes, métodos, etc

Introdução à programação

- No Python todo identificador de começar por uma letra, ou o símbolo '_' e ser seguido letras, dígitos ou '_'

identifier ::= (letter | "_") (letter | digit | "_")*

letter ::= lowercase | uppercase

lowercase ::= "a"..."z"

uppercase ::= "A"..."Z"

digit ::= "0"..."9"

Introdução à programação

- Vejamos exemplos de expressões
 - Aritméticas: $(5 + 3)$, $(a + 2)$, $(a**b+(3-4))$
 - Lógicas: $(a \text{ or } b)$, $(a \text{ and } b)$, $(\text{not}(a) \text{ ou } b)$
 - Relacionais $(a \geq c)$, $(a \neq b)$
 - Bitwise: $a \ll 2$, $a \gg 2$, $((a \gg 2) \ll 4)$
- Observem que as expressões combinam segundo uma regra, literais, variáveis e constantes com operadores compatíveis para os tipos usados
- As expressões também podem ser parentizadas

Introdução à programação

- Um dos aspectos fundamentais é obter os dados. Isto é feito via operações de Entrada e Saída:
 - É toda a comunicação entre o usuário e o computador
 - Em Python, ocorre através dos comandos print e input

```
print (“Nome do jogo”) #nome provisório
```

```
nome = input("Digite o seu nome.")
```

```
idade = input("Digite sua idade.")
```

```
print(“A única coisa da qual me lembrava era que me chamava” +  
nome)
```

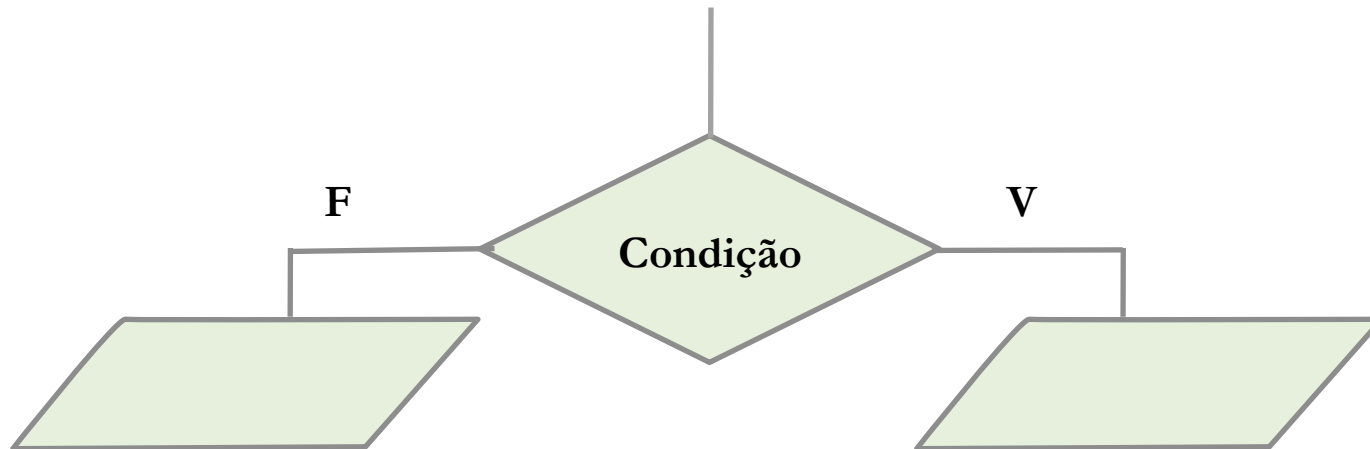
Introdução à programação

- Desvios condicionais



Introdução à programação

- Para mudar o fluxo de execução dos comandos de um programa usamos desvios condicionais



Introdução à programação

- Sua sintaxe é:

if(condição 1):

O que fazer se condição 1 for verdadeira

elif(condição 2):

O que fazer se condição 1 for verdadeira e condição 2 falsa

else:

Ação ao não ocorrer nenhuma das duas

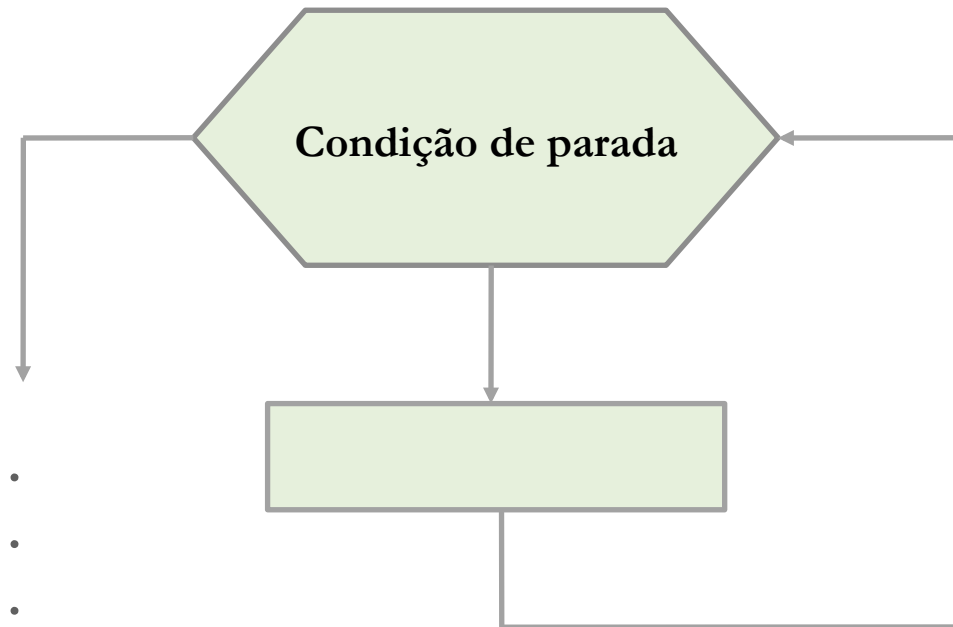
Introdução à programação

- Repetições



Introdução à programação

- Repetições: a necessidade de realizar a mesma tarefa repetidas vezes é constante no meio da programação, para evitar repetir código utilizamos de ESTRUTURAS DE REPETIÇÃO



Introdução à programação

- Em python temos duas estruturas de repetição:
 - while : repetição de acordo com uma condição
 - for : repetição por um número determinado de vezes

Introdução à programação

- A sintaxe do comando while é:

while(condição):

o que se repetir enquanto a condição for verdadeira

Introdução à programação

- A sintaxe do comando for é:

for variável **in** conjunto:

o que fazer para cada valor da
variável no conjunto

Introdução à programação

- Cadeias de caracteres: Strings
- Cadeias de caracteres

Introdução à programação

- Variáveis compostas



Introdução à programação

- Variáveis compostas:
- Ao criarmos programas mais complexos precisamos criar uma grande quantidade de variáveis, porém, às vezes, não sabemos previamente quantas
- Para isso utilizamos VARIÁVEIS COMPOSTAS.

Introdução à programação

- Na linguagem Python, uma variável composta é expressa na forma de um lista
- Um lista é definida através de uma sequência de valores separados por vírgula e delimitados por abre e fecha colchetes

Introdução à programação

- Cada valor em uma lista ocupa uma posição definida por um número inteiro denominado índice
- É possível consultar ou modificar um valor da lista através da informação de seu índice

Introdução à programação

- Criação de uma lista

```
nomeDaLista = [valor1, valor2, valor3, ...]
```

- Acesso a uma posição

```
nomeDaLista[posição]
```

Introdução à programação

- Existe outras operações:

Inserir um elemento no início

```
nomeDaLista.append(valorK)
```

Remover do início

```
ultimoElemento = nomeDaLista.pop()
```

Remover do final

```
primeiroElemento = nomeDaLista.remove()
```

Introdução à programação

- Para construir um jogo, precisamos programá-lo de forma que possamos fazer o computador:
 - Descrever a ambientação do jogo
 - Exibir os gráficos
 - Implementar as regras do jogo,
 - Reagir aos comandos realizados no teclado, mouse ou personagens
 - Simular o ambiente do jogo incluindo efeitos simplificados de física
 - Expor um nível de inteligência nos inimigos e desafios
 - Reavaliar as condições e estados do jogo
 - Realizar as animações
 - Etc...
- Nesta aula começamos apenas o primeiro passo para atingir tal objetivo

Introdução à programação

- Exercício: Implemente um jogo simples em que o jogador deve adivinhar um número gerado pelo computador.
- Regras:
 - O computador sorteia um número de 1 a 10 e o jogador tem 3 chances de advinha-lo
 - Caso erre, o computador indica se o jogador indicou um número menor ou maior que o por ele escolhido
 - Se o jogador adivinhar um número ele ganha 1 ponto, caso contrário ganha 1 ponto o computador
 - O jogo é composto de várias rodadas
 - Ganha o jogo (computador ou usuário) quem tiver feito primeiramente superado o adversário por 2 pontos

Referências

- Geral:

Material sobre Introdução à Programação em Python - Jovens Talentos da UFF – 2012

- Operadores em Python

http://www.tutorialspoint.com/python/python_basic_operators.htm