

LINGUAGENS DE PROGRAMAÇÃO

Maria Inês Castilho

LINGUAGEM DE PROGRAMAÇÃO PYTHON



LINGUAGEM DE PROGRAMAÇÃO C

```
#include <stdio.h>
#include <iso646.h>

void main(){

    int a = 1, b=2;

    if(a==0 and b==0)
        printf("A e B sao zero");
    else if (a==1 or b==1)
        printf("Pelo menos um valor eh um");
    else if (a==2 && b==2)
        printf("A e B sao iguais a dois");
}
```



LINGUAGEM DE PROGRAMAÇÃO **JAVA**

```
4 var seconds = now.getSeconds();
5
6 var ampm = "am";
7 var colon = '<IMG SRC="images/colon.gif">';
8
9 if (hours >= 12) {
10     ampm = "pm";
11     hours = hours - 12;
12 }
13
14 if (hours == 0) hours = 12;
15
16 if (hours < 10) hours = "0" + hours;
17 else hours = hours + '';
18
19 if (minutes < 10) minutes = "0" + minutes;
20 else minutes = minutes + '';
21
```

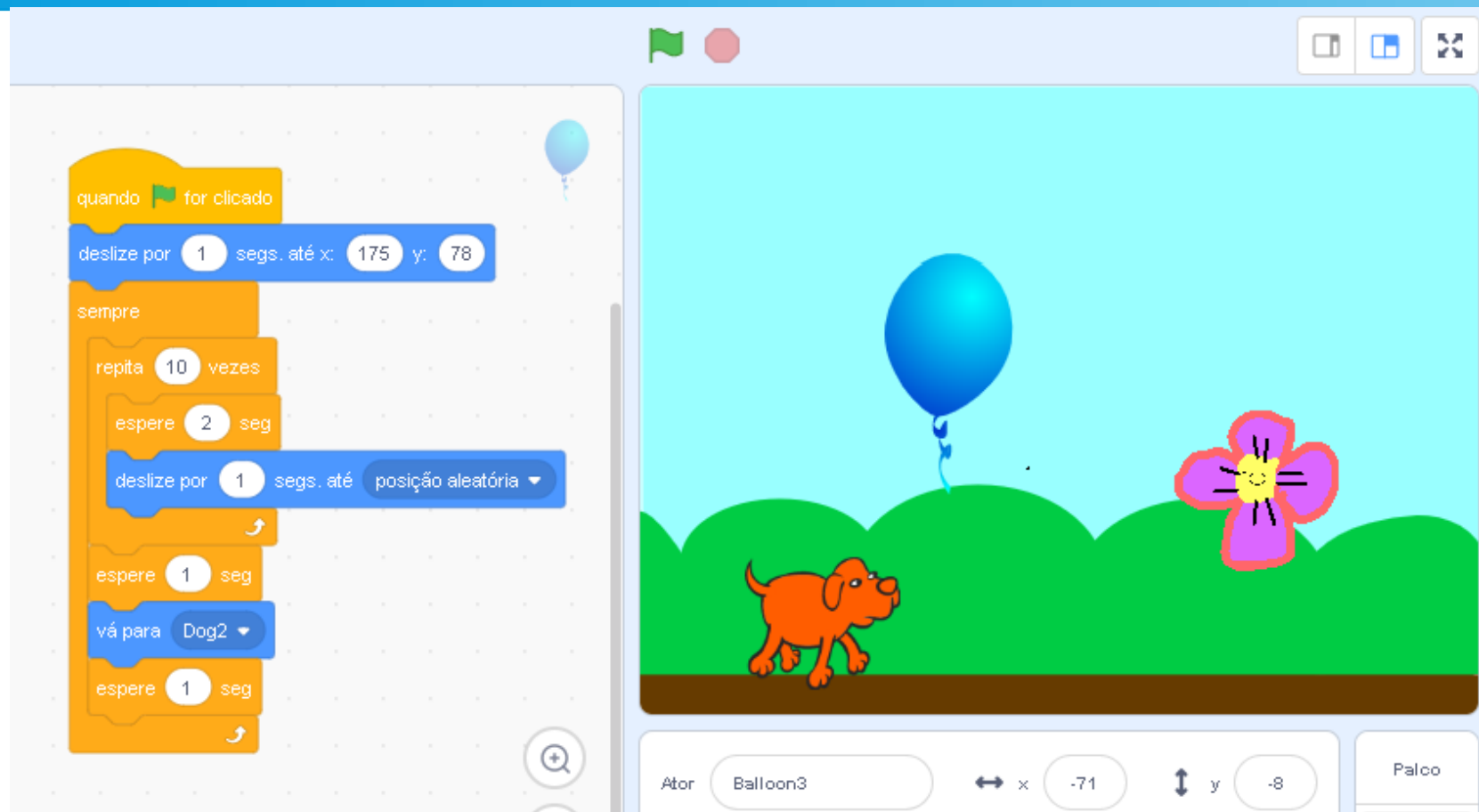


LINGUAGEM DE PROGRAMAÇÃO JAVASCRIPT

```
funcao01.js JS funcao02.js JS funcao03.js JS funcao04.js JS funcao05.js
1 // RECURSIVIDADE
2 function fatorial(n) {
3     if (n == 1) {
4         return 1
5     } else {
6         return n * fatorial(n-1)
7     }
8 }
9
10
11
12 /*
13
14 5! = 5 x 4 x 3 x 2 x 1
15 5! = 5 x 4!
16
17 n! = n x (n-1)!
18 1! = 1
19
20 */
```

A yellow square containing the letters 'JS' in a bold, black, sans-serif font.

LINGUAGENS DE PROGRAMAÇÃO SCRATCH



The image displays the Scratch programming environment. On the left, the code editor shows a script for a balloon character:

- quando for clicado
- deslize por 1 segs. até x: 175 y: 78
- sempre
 - repita 10 vezes
 - espere 2 seg
 - deslize por 1 segs. até posição aleatória
 - espere 1 seg
 - vá para Dog2
 - espere 1 seg

On the right, the stage shows a blue balloon on a green hill. The stage properties at the bottom indicate the balloon's current position: x: -71, y: -8.

HTML

Usada para criar páginas para a internet não é uma linguagem de programação propriamente dita. É sim uma linguagem de marcação. O nome já diz isso: HTML = Hyper Text Markup Language

No entanto, associada ao SQL , que é uma linguagem de consulta de banco de dados, torna-se muito eficiente em formatar páginas web.

NOTA: SQL = Structured Query Language = Linguagem de Consulta Estruturada

```
1 <section>
2   <h2>Tabular Tables</h2>
3   <table>
4     <thead>
5       <tr>
6         <th>Item</th>
7         <th>Value</th>
8         <th>Quantity</th>
9       </tr>
10    </thead>
11    <tbody>
12      <tr>
13        <td>Apples</td>
14        <td>$1</td>
15        <td>7</td>
16      </tr>
17      <tr>
18        <td>Lemonade</td>
19        <td>$2</td>
20        <td>18</td>
21      </tr>
22      <tr>
23        <td>Hand Sanitizer</td>
24        <td>$999</td>
25        <td>2</td>
26      </tr>
```

ALGORÍTMO

Algoritmo é uma sequência de instruções para que se obtenha um determinado objetivo, sendo que os passos devem ser finitos, executáveis e ordenados na sequência correta.

É normal que qualquer pessoa já tenha feito um algoritmo de forma simplificada (ou em pensamento), como por exemplo:

- ~trajeto de ir da escola para casa,
- ~sequência de passos para acessar um filme na Netflix,
- ~arrumar a cama, etc
- ~dizer para o computador executar operação adição entre dois números, sem usar softwares específicos.

Um programador profissional segue etapas muito definidas. Se você quiser saber mais sobre algoritmos aconselho o site:

Ada Lovelace



Augusta Ada Byron King, Condessa de Lovelace (10/12/1815 — 27 /11/ 1852), atualmente conhecida como **Ada Lovelace**.

Essa matemática e escritora inglesa foi quem **primeiro escreveu um algoritmo** que consistia numa sequência de passos para resolver um cálculo matemático. **Hoje é reconhecido como o primeiro programa de computador**

O que uma linguagem de programação deve fazer para ser considerada completa?

- ~fazer cálculos;
- ~mudar informações contidas em algum tipo de memória;
- ~tomar decisões;
- ~mudar o fluxo de execução.


Estes são atributos da Máquina de Turing

Alan Turing (1912-1954) criou uma máquina teórica que permitia fazer operações específicas. Mais tarde, passou-se a considerar que qualquer linguagem de programação precisa ser capaz de fazer todas as operações desta máquina universal.



Exemplo de uma estrutura de programação e o resultado visível do produto final

```
1  /*      *      *      *      *      *      *
2  . * move your mouse to over the stars .
3  * . . change these values: . *
4  *      *      *      *      *      */
5  const STAR_COLOR = '#fff';
6  const STAR_SIZE = 3;
7  const STAR_MIN_SCALE = 0.2;
8  const OVERFLOW_THRESHOLD = 50;
9  const STAR_COUNT = ( window.innerWidth + window.innerHeight ) / 8;
10
11  const canvas = document.querySelector( 'canvas' ),
12  context = canvas.getContext( '2d' );
13
14  let scale = 1, // device pixel ratio
15  width,
16  height;
17
18  let stars = [];
19
20  let pointerX,
21  pointerY;
22
23  let velocity = { x: 0, y: 0, tx: 0, ty: 0, z: 0.0005 };
24
25  let touchInput = false;
26
27  generate();
28  resize();
29  step();
30
31  window.onresize = resize;
```



Veja animado em:

<https://codepen.io/hakimel/embed/bzrZGo?height=300&theme-id=37007&default-tab=js,result&editable=true>

Existe uma melhor linguagem de programação?

Não!

Cada linguagem tem sua finalidade.

No entanto, elas podem ser classificadas em linguagem de

ALTO NÍVEL e de **BAIXO NÍVEL**

linguagem de alto nível		linguagem de baixo nível	
PRÓS	CONTRAS	PRÓS	CONTRAS
<ul style="list-style-type: none">facilidade de aprendizagem;produtividade;permite a escrita de mais códigos em menos tempo;fácil entendimento;manutenção simplificada;baixo custo operacional.	<ul style="list-style-type: none">desempenho de um programa pode ser prejudicado, pois ele exige maior tempo de processamento;ocupa mais memória quando comparadas a uma linguagem de baixo nível.	<ul style="list-style-type: none">tempo de processamento mais rápido do que o de uma de alto nível;melhor aproveitamento da arquitetura do computador.	<ul style="list-style-type: none">maior tempo para compreender e dominar a sintaxe;necessidade de conhecer profundamente o hardware da máquina, o que exige mais investimento em estudo e treinamento.

POR QUE TODOS DEVEM APRENDER A PROGRAMAR?

- Estimula o raciocínio lógico.
- Ajuda na organização de modo geral.
- Propicia uma melhor escrita.
- Incentiva o aprendizado de matemática, física e língua inglesa.
- Auxilia no desempenho pessoal e profissional.
- Estimula a criatividade.
- Desenvolve habilidades para solucionar situações adversas.

BENEFÍCIOS DA APRENDIZAGEM DE PROGRAMAÇÃO EM CRIANÇAS E ADOLESCENTES

Estimula o raciocínio lógico

Ao aprender a linguagem de programação, os jovens se habitua a pensar de forma estruturada. Eles designam ações a serem cumpridas pelo computador, através de códigos específicos, que são criados por sequências de números e palavras. Desse modo, desenvolve o lado esquerdo do cérebro, responsável pelo raciocínio lógico, analítico e crítico.

Ajuda na organização de modo geral

O aprendizado da programação estimula o jovem a organizar os pensamentos e as ações a serem tomadas para a solução dos desafios envolvidos nos projetos de criação de um game ou aplicativo, por exemplo. Isso acaba refletindo na capacidade de organização de um modo geral, como no planejamento de tarefas e atividades, estruturação de pensamentos e até mesmo na maneira de estudar.

Propicia uma melhor escrita

O ensino de programação auxilia e melhora o aprendizado de matérias de ciências humanas, além de melhorar a escrita. Uma vez que o jovem aprende a organizar melhor suas ideias e pensamentos, ele consegue estruturar com mais facilidade o texto a ser escrito.

Incentiva o aprendizado de Matemática, Física e Língua Inglesa

Coordenadores, professores e pais vêm percebendo grandes melhoras no desempenho escolar em diversas disciplinas escolares, principalmente, nas matérias que tem como base o raciocínio lógico, como a matemática, física e também o inglês. As crianças e adolescentes se familiarizam com os números ou as novas palavras e aprendem raciocinar com mais precisão, entendendo a teoria e conseguindo aplicá-la na prática.

Auxilia no desempenho pessoal e profissional

Aprender a programar auxilia a criança e o adolescente a descobrir suas potencialidades e estimular suas aptidões. Eles se tornam mais engajados e entusiasmados a seguir em busca de novos desafios. Além disso, no futuro, por terem habilidades diferenciadas, deverão se destacar no mercado de trabalho.

Estimula a criatividade

Através da criação de animações, games e apps, os jovens aprendem a pensar de forma estruturada e não apenas a decorar fórmulas ou datas. Na programação, é indispensável o processo de analisar, planejar, criar e executar um projeto valorizando todas as suas etapas. Além de incentivar a criatividade, estimulam as crianças e os adolescentes a trabalharem em equipe, visto que é mais fácil trabalhar e achar soluções quando se discute o assunto com uma ou duas pessoas.

Desenvolve habilidades para solucionar situações adversas

Outro benefício que podemos destacar no aprendizado de programação é a capacidade dos estudantes de solucionar problemas. No mundo da tecnologia, para conseguir se comunicar com eficiência, certos padrões lógicos devem ser seguidos. Se, por acaso, uma sequência de códigos não for desenvolvida corretamente, o projeto não responderá de acordo, então o aluno se depara com uma situação que precisará ser resolvida, a fim de cumprir o desafio proposto.

*Thank
you*

