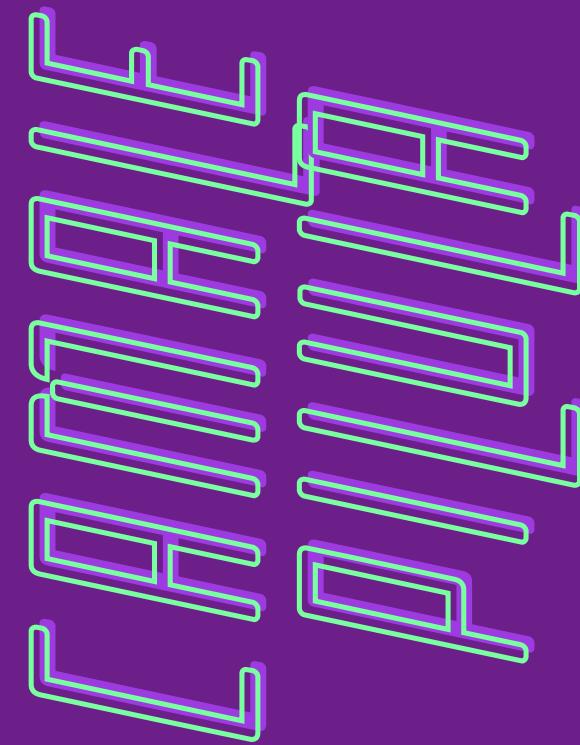
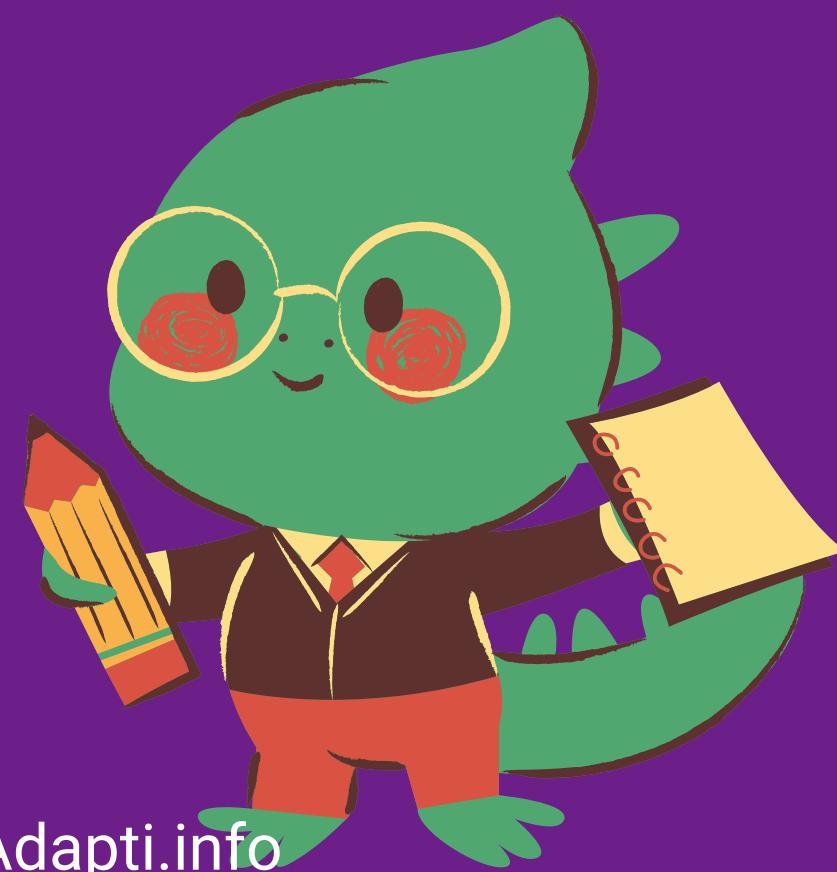


x

ADAPTELAS

SEGUNDO DIA!

Tema: Experiência no mundo da programação



Ana Flávia Frontino da Cruz

Backend Developer no Picpay



RETROSPECTIVA COM DIQUINHAS

- Inicio 2015/02
- Adulti 2020 - 2021
- Estágio - Giro Carteiras
- Avanzzada
- Picpay



⋮

**Em uma
escala
de 1 a 9, como
você está
hoje?**



Agradecemos pelos feedbacks!



VAMOS RELEMBRAR NOSSO OBJETIVO FINAL!

Seguimos juntas!



ADAPT Elas

Página 1 Página 2 Página 3

Blog



Quem sou eu?



Lore ipsum dolor sit amet, consectetur adipiscing elit. Nam aliquam risus ornare ultrices.

John Doe Jan 1, 2020 12 Comments

Lore ipsum dolor sit amet, consectetur adipiscing elit. Sed pellentesque nulla felis, sit. Lore ipsum dolor sit amet, Lore ipsum dolor sit amet, consectetur adipiscing elit. Sed pellentesque nulla felis, sit. Lore ipsum dolor sit amet.

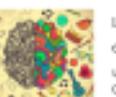
[Ver mais →](#)



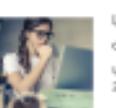
Categorias

- Lore ipsum dolor sit
- Lore ipsum dolor sit
- Lore ipsum dolor sit

Posts recentes



Lore ipsum dolor sit amet, consectetur adipiscing elit. Nam aliquam risus ornare ultrices. 01/01/2021



Lore ipsum dolor sit amet, consectetur adipiscing elit. Nam aliquam risus ornare ultrices. 25/04/2021

Tags

Todos Lore ipsum Lore ipsum

Vamos dar
inicio a nossa
aula!



Dúvidas da Atividade 1 ou da Aula 1



O que vamos estudar hoje?

- Condicionais
- Operadores lógicos
- Laços de repetição
- Arrays
- Duvidas



Condicionais

A Estrutura Condicional Simples executa um comando ou vários comandos se a condição for verdadeira. Se a condição for falsa, a estrutura é finalizada sem executar os comandos.

Condicionais



Portugol

```
se (condicao)
{
    execute isso
}
senao
{
    execute outra coisa
}
```



Outras linguagens

```
if (condicao)
{
    execute isso
}
else
{
    execute outra coisa
}
```

**Mas como são as
condições que a gente
pode colocar dentro
dessas parênteses?**



Operadores de Relação

Operador	Nome	Exemplo
<code>==</code>	Igual	<code>\$a == \$b</code>
<code>!=</code>	Diferente	<code>\$a != \$b</code>
<code><></code>	Diferente	<code>\$a <> \$b</code>
<code><</code>	Menor que	<code>\$a < \$b</code>
<code>></code>	Maior que	<code>\$a > \$b</code>
<code><=</code>	Menor ou igual	<code>\$a <= \$b</code>
<code>>=</code>	Maior ou igual	<code>\$a >= \$b</code>

Vamos para um exemplo!

Maioridade Penal

<https://portugol-webstudio.cubos.io/ide#share=68320>

```
programa
{
    funcao inicio()
    {
        const inteiro MAIORIDADE = 18

        inteiro idade, anos

        escreva("Digite sua idade: ")
        leia(idade)

        // Calcula quantos anos faltam para atingir a maioridade
        anos = MAIORIDADE - idade

        se (anos > 0)
        {
            escreva("Falta(m) ", anos, " ano(s) para você atingir a maioridade\n")
        }
        senao
        {
            escreva("Você já atingiu a maioridade\n")
        }
    }
}
```

Mas e se eu quiser conferir outras condições ao mesmo tempo?

Operadores Lógicos

Operador	Significado
&&	AND
	OR
!	NOT

E
OU
NÃO

Vamos para um exemplo!

Vogais

<https://portugol-webstudio.cubos.io/ide#share=68327>

```
programa
{
    funcao inicio ()
    {
        caracter letra

        escreva("Digite uma letra: ")
        leia(letra)

        // O Portugol diferencia caracteres minúsculos e maiúsculos,
        // portanto é preciso verificar ambos os casos

        se
        (
            letra == 'A' ou letra == 'E' ou letra == 'I' ou letra == 'O' ou letra == 'U' ou
            letra == 'a' ou letra == 'e' ou letra == 'i' ou letra == 'o' ou letra == 'u'
        )
        {
            escreva("\nA letra '", letra, "' é uma vogal\n")
        }
        senao
        {
            escreva("\nA letra '", letra, "' é uma consoante\n")
        }
    }
}
```

Em outras linguagens...

A screenshot of a terminal window. The code in the editor pane is:

```
if (nota_final >= 6.0 && frequencia >= 75 )
    printf("Aluno aprovado \n");
else
    printf("Aluno reprovado \n");
```

The terminal window shows the following interaction:

```
Digite a nota final do aluno: 7
Informe a frequencia do aluno: 75
Aluno aprovado
```

A screenshot of a terminal window. The code in the editor pane is:

```
if (temperatura > limite_max || temperatura < limite_min)
    printf("Alerta: Temperatura fora da faixa permitida!!! \n");
else
    printf("Temperatura OK \n");
```

The terminal window shows the following interaction:

```
Informe o valor da temperatura: 7
Temperatura OK
```

Laços de repetição

Laço FOR

Usamos o laço **for** quando sabemos exatamente **quantas vezes queremos repetir um bloco de instruções**.

```
for (valor inicial; teste booleano; incremento)  
{  
    //Lógica  
}
```

Vamos para um exemplo!

Soma

<https://portugol-webstudio.cubos.io/ide#share=68504>

```
programa
{
    funcao inicio()
    {
        inteiro soma = 0, numero, contador

        escreva("Digite o número até o qual deseja somar: ")
        leia(numero)

        // Repete até o contador atingir o valor informado pelo usuário

        para (contador = 0; contador <= numero; contador++)
        {
            soma = soma + contador // Soma o valor atual do contador
        }
        escreva("A soma de 1 até ", numero, " é: ", soma, "\n")
    }
}
```

Em outras linguagens...

```
for (int a = 0; a < 5; a++)
```

```
{
```

**aqui seu código
que você quer
que repita**

```
}
```

Laços de repetição

Laço WHILE

Esta instrução é usada quando não sabemos quantas vezes um determinado bloco de instruções precisa ser repetido. Com ele, a execução das instruções vai continuar até que uma condição seja verdadeira.

```
while (teste condicional)
{
    //comandos; > serão executados enquanto o teste condicional for igual a verdadeiro (true)
}
```

Vamos para um exemplo!

Verificação

<https://portugol-webstudio.cubos.io/ide#share=68506>

```
programa
{
    funcao inicio()
    {
        inteiro idade

        faca
        {
            escreva ("Informe sua idade (valores aceitos de 5 a 150): ")
            leia (idade)
        }
        enquanto (idade < 5 ou idade > 150)

        // A partir deste ponto do código é garantido que a idade
        // terá um valor válido e não causará erros inesperados

        escreva ("\nCorreto!\n")
    }
}
```

Arrays

Pense no seguinte problema...

Você precisa criar um algoritmo que lê o nome e as 4 notas de 50 alunos, calcular a média de cada aluno e informar quais foram aprovados e quais foram reprovados.

Se a gente fosse criar uma variável para cada informação dessa, seriam **50 variáveis para armazenar os nomes dos alunos**, ($4 * 50 =$) **200 variáveis para armazenar as 4 notas de cada aluno** e por fim, **50 variáveis para armazenar as médias de cada aluno**.

Arrays



Isso tem solução? A gente consegue armazenar várias variáveis do mesmo tipo??

Arrays

A principal finalidade do array é agrupar uma coleção de elementos do mesmo tipo em uma única variável, ou seja, para cada linha de programação, um dado.

Vetor de nomes dos alunos

0	1	2	...	48	49
João	Pedro	Carlos	...	José	Maria

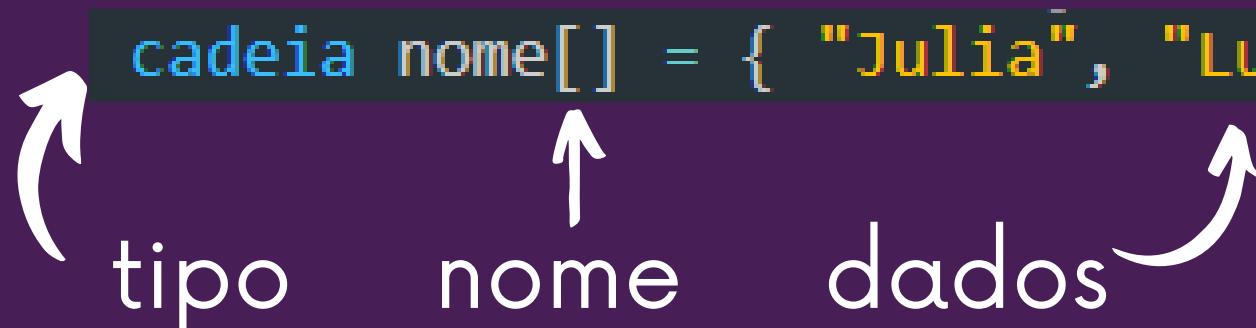
Matriz das notas dos alunos

0	1	2	3	
0	9,5	10	8	7,5
1	10	9	9	5,5
2	9	8,5	9,5	7
...
48	7	10	10	9
49	7	8,5	5,5	4

Arrays

SINTAXE em Portugol

```
cadeia nome[] = { "Julia", "Luisa" , "Amanda", "Carla", "Ana" }
```



SINTAXE em PHP

```
$media =array[13,15,23];
```



isso aqui tem que ter

Vamos para um exemplo!

Tabela de Dados

<https://portugol-webstudio.cubos.io/ide#share=68523>

```
programa
{
    funcao inicio()
    {
        // Criação dos vetores, já com os dados inicializados
        cadeia nome[] = { "Julia", "Luisa", "Amanda", "Carla", "Ana" }
        real altura[] = { 1.71, 1.78, 1.75, 1.87, 1.71 }

        // Cria o cabeçalho da tabela
        escreva ("-----\n")
        escreva ("      TABELA      \n")
        escreva ("-----\n")

        /* Percorre os vetores exibindo as informações. Note que as informações
         * são relacionadas, colocando-as na mesmas na mesma posição em ambos os
         * vetores. Por exemplo, se quiséssemos incluir a idade de cada pessoa,
         * criaríamos mais um vetor do tipo inteiro e a idade de Andre, seria
         * armazenada na posição 0 do novo vetor, a idade de Thiago, na posição 1,
         * a de Bruno na posição 2, e assim por diante.
         */
        para (inteiro posicao = 0; posicao < 5; posicao++)
        {
            // O caracter especial \t serve para escrever uma tabulação
            escreva (nome[posicao], "\t\t", altura [posicao], "\n")
        }
    }
}
```

Ufa! Terminamos por hoje...



Até a próxima aula!
Telegram e Classroom
aberto para dúvidas
e sugestões!