

# Computação II - Python

## Aula 1 - Apresentação do curso e revisão de programação modular

Carla A. D. M. Delgado

João C. P. da Silva

Dept. Ciência da Computação - UFRJ

# De onde viemos...

Em computação 1 abordamos **programação estruturada** e os **conceitos básicos** de programação em Python:

- tipos de dados simples e compostos
- operações e expressões com dados
- comandos da linguagem de blocos de comandos
- estrutura condicional
- estruturas de repetição
- **estilo modular de programação** (organizado em funções)
- interação linear com o usuário usando o console.

# Onde estamos...

O conceito central e que define nosso atual estilo de programação era a **Função**. Somos capazes de

- projetar e construir funções que executam tarefas simples
- projetar programas maiores, articulando o uso de diversas funções para resolver problemas não tão simples
- fazer uso de funções previamente implementadas por nós mesmos ou outras pessoas através da importação de módulos
- testar nossas funções e programas
- construir programas legíveis, organizados, confiáveis (afinal, sabemos testar e avaliar o que programamos), mas com baixo grau de sofisticação.

# Pra onde vamos...

## Neste módulo

- serão abordados conceitos mais sofisticados de programação, que permitirão interagir com o usuário de modo não-linear, usando uma interface gráfica, bem como trabalhar com conjuntos de dados e usar Python para computação científica.
- objetivo (ambicioso!): apropriar-se da programação como ferramenta de trabalho
- ser capaz de projetar e construir programas que sejam úteis em sua rotina profissional.

# Modularização

Faça um programa que dado o salário bruto calcula o salário líquido. O salário líquido é calculado a partir do salário bruto, primeiro descontando 11% referente ao INSS, e do resultado, descontando-se 15% de imposto de renda (IR).

## Exemplo

- Salário Bruto = R\$ 5000,00
- Desconto do INSS = R\$ 550,00 (11% de R\$ 5000,00)
- Desconto do IR = R\$ 667,50 (15% de R\$ 4450,00)
- Salário Líquido =  $5000 - (550 + 667,50) = 3782,50$

# Modularização

Faça um programa que dado o salário bruto calcula o salário líquido. O salário líquido é calculado a partir do salário bruto, primeiro descontando 11% referente ao INSS, e do resultado, descontando-se 15% de imposto de renda (IR).

```
1 def salarioLiquido(salarioBruto):
2     """ Dado o salario bruto calcula o salario liquido
3     Entrada: float
4     Saida: float """
5
6     descontoINSS = (salarioBruto)*0.11
7     descontoIR = (salarioBruto - descontoINSS)*0.15
8     salarioLiq = (salarioBruto - descontoINSS - descontoIR)
9     return salarioLiq
```

# Modularização

Faça um programa que dado o salário bruto calcula o salário líquido. O salário líquido é calculado a partir do salário bruto, primeiro descontando 11% referente ao INSS, e do resultado, descontando-se 15% de imposto de renda (IR).

```
1 def salarioLiquido(salarioBruto):
2     """ Dado o salario bruto calcula o salario liquido
3     Entrada: float
4     Saida: float """
5
6     descontoINSS = (salarioBruto)*0.11
7     descontoIR = (salarioBruto - descontoINSS)*0.15
8     salarioLiq = (salarioBruto - descontoINSS - descontoIR)
9     return salarioLiq
```

Note que o cálculo do desconto é sempre feito de uma mesma maneira.  
Podemos então generalizar seu cálculo.

# Modularização

Faça um programa que dado o salário bruto calcula o salário líquido. O salário líquido é calculado a partir do salário bruto, primeiro descontando 11% referente ao INSS, e do resultado, descontando-se 15% de imposto de renda (IR).

```
1 def salarioLiquido(salarioBruto):
2     """ Dado o salario bruto calcula o salario liquido
3     Entrada: float
4     Saida: float """
5
6     descontoINSS = (salarioBruto)*0.11
7     descontoIR = (salarioBruto - descontoINSS)*0.15
8     salarioLiq = (salarioBruto - descontoINSS - descontoIR)
9     return salarioLiq
```

Note que o cálculo do desconto é sempre feito de uma mesma maneira.  
Podemos então generalizar seu cálculo.

```
1 def calculadesconto(valorBruto, percentual):
2     """ Dado um valor bruto e um percentual de desconto
3     calcula o valor com o desconto aplicado
4     Entrada: float, float
5     Saida: float """
6
7     desconto = valorBruto*percentual/100.0
8     valorLiquido = valorBruto-desconto
9     return valorLiquido
```



# Modularização

Faça um programa que dado o salário bruto calcula o salário líquido. O salário líquido é calculado a partir do salário bruto, primeiro descontando 11% referente ao INSS, e do resultado, descontando-se 15% de imposto de renda (IR).

```
1 def calculadesconto(valorBruto, percentual):
2     """Dado um valor bruto e um percentual de desconto
3     calcula o valor com o desconto aplicado
4     Entrada: float, float
5     Saida: float"""
6
7     desconto = valorBruto*percentual/100.0
8     valorLiquido = valorBruto-desconto
9     return valorLiquido
```

Usando a função acima:

# Modularização

Faça um programa que dado o salário bruto calcula o salário líquido. O salário líquido é calculado a partir do salário bruto, primeiro descontando 11% referente ao INSS, e do resultado, descontando-se 15% de imposto de renda (IR).

```
1 def calculadesconto(valorBruto, percentual):
2     """Dado um valor bruto e um percentual de desconto
3     calcula o valor com o desconto aplicado
4     Entrada: float, float
5     Saida: float"""
6
7     desconto = valorBruto*percentual/100.0
8     valorLiquido = valorBruto-desconto
9     return valorLiquido
```

Usando a função acima:

```
1 def salarioLiquido(salarioBruto):
2     """Dado o salario bruto calcula o salario liquido
3     Entrada: float
4     Saida: float"""
5
6     salariosemINSS = calculadesconto(salarioBruto, 11.0)
7     salariosemIRsemINSS = calculadesconto(salariosemINSS, 15.0)
8     return salariosemIRsemINSS
```

# Modularização

- Torna o código mais legível.
- Permite que algumas funcionalidades sejam reaproveitadas.
- Permite que partes do código sejam testadas isoladamente.

# Modularização

- Torna o código mais legível.
- Permite que algumas funcionalidades sejam reaproveitadas.
- Permite que partes do código sejam testadas isoladamente.

```
1 def calculadesconto(valorBruto, percentual):  
2     """Dado um valor bruto e um percentual de desconto  
3     calcula o valor com o desconto aplicado  
4     Entrada: float, float  
5     Saída: float"""  
6  
7     desconto = valorBruto*percentual/100.0  
8     valorLiquido = valorBruto-desconto  
9     return valorLiquido
```

```
1 In [1]: calculadesconto(5000.00,11.0)  
2 Out[1]: 4450.0  
3  
4 In [2]: calculadesconto(4450.00,15.0)  
5 Out[2]: 3782.5
```

# Modularização

- A programação permeia muitas áreas atualmente. O nível de sofisticação esperado é elevado.
- Para justificar o investimento, um trecho de código deve ser reutilizável. Várias pessoas reutilizam e compartilham seus códigos, criando comunidades.
- A **organização e a legibilidade** do código são requisitos essenciais, tão importantes quanto eficiência e eficácia.
- Ter seu código bem documentado também é importante para que você mesmo consiga utilizá-lo futuramente. É muito fácil esquecer o que um código faz, e gasta-se tempo para tentar entendê-lo novamente.

# Modularização

- Para atender às demandas atuais, cada vez mais complexas e sofisticadas, o programador deve pesquisar sobre bibliotecas disponíveis que o auxiliem.
- Tão importante quanto saber programar tudo o que precisa, é saber aproveitar o que está disponível, ou não é possível atender às demandas em tempo aceitável.
- É importante ser capaz de ler e entender a documentação de tais bibliotecas. Um bom conhecimento da linguagem de programação em que a biblioteca foi escrita ajuda muito.
- Uma vez que se produza código reutilizável e de boa qualidade, é de bom tom compartilhar! A comunidade agradece e reconhece o esforço.

# Modularização

Vamos fazer o programa chamado *Campeonato*. O programa deve:

- possuir um menu principal com o seguinte layout:

```
Escolha uma opção :  
1 - informar o resultado e estatísticas de uma partida  
2 - obter informacoes sobre uma partida  
3 - classificação e saldo de gols  
0 - terminar  
Opção :
```

- possuir uma tela para lançamento dos resultados dos jogos com o seguinte layout:

```
Jogo (escreva fim para terminar) : Brasil x Bélgica  
Resultado : 1 x 2  
Numero de passes certos - Brasil : 80  
Numero de passes errados - Brasil : 25  
Numero de passes certos - Bélgica : 120  
Numero de passes errados - Bélgica : 35  
  
Jogo (escreva fim para terminar) : fim
```

- os dados fornecidos nesta tela devem ser armazenado em uma matriz da seguinte forma:  
 `[['Brasil',1,80,25,'Bélgica',2,120,35]]`

# Modularização

Vamos fazer o programa chamado *Campeonato*. O programa deve:

- possuir um menu para escolher qual partida terá suas estatísticas apresentadas com o seguinte layout:

```
Escolha uma partida:  
1- 'Brasil x Belgica'  
2- 'Brasil x Alemanha'  
0- terminar  
Opção :
```

- possuir uma saída das estatísticas de uma partida com o seguinte layout:

```
Partida - Brasil x Belgica  
Resultado: 1 x 2  
Total de passes - Brasil: 105 (80 certos e 25 errados)  
Total de passes - Belgica: 155 (120 certos e 35 errados)
```



# Modularização

Vamos fazer o programa chamado *Campeonato*. O programa deve:

- mostrar a classificação com o saldo de gols. Lembre que a vitória dá 3 pontos, o empate 1 para cada time e a derrota não dá pontos.

## Exemplo

Classificacao	Pontos	SG
1 - Alemanha	3	6
2 - Belgica	3	1
3 - Brasil	0	-7

# Computação II - Python

## Aula 1 - Apresentação do curso e revisão de programação modular

Carla A. D. M. Delgado

João C. P. da Silva

Dept. Ciência da Computação - UFRJ