Recordação

- Variáveis e entrada de dados
- Condições
- Repetições

Variáveis

Entrada de Dados

Condições

Condições

Repetições

Repetições

Listas

Lucas Gonçalves Nadalete

lucas.nadalete@fatec.sp.gov.br

Material produzido pelo Prof. Masanori

Edifício

- Edifício de apartamentos

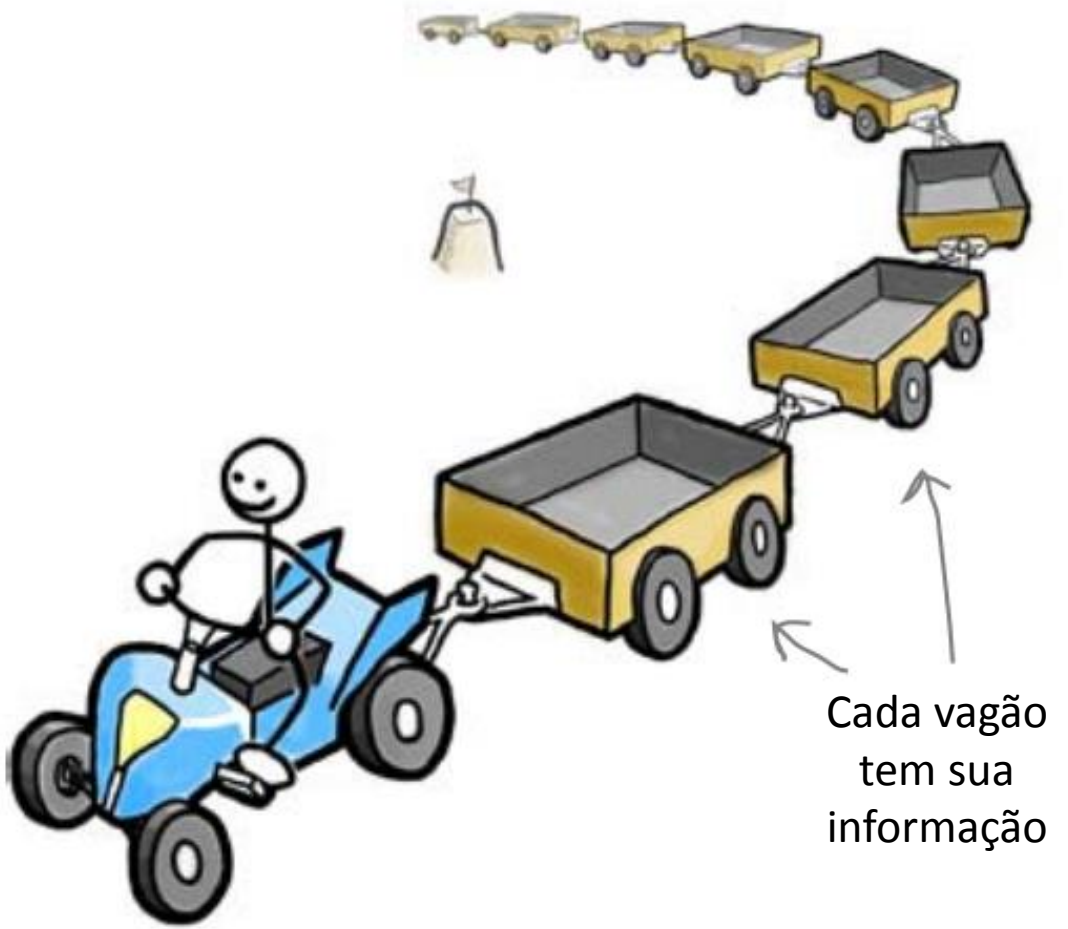
```
edifício_térreo   = "Família Souza"
edifício_1o_andar = "Família Brito"
edifício_2o_andar = "Sr Jorge"
edifício_3o_andar = "Família Tanaka"
```

Edifício

- Podemos associar o térreo ao andar zero, o primeiro é o andar 1 e assim por diante

```
edifício = ["Família Souza",
            "Família Brito",
            "Sr Jorge",
            "Família Tanaka"]
print (edifício[0])
print (edifício[1])
print (edifício[2])
print (edifício[3])
>>>
Família Souza
Família Brito
Sr Jorge
Família Tanaka
```

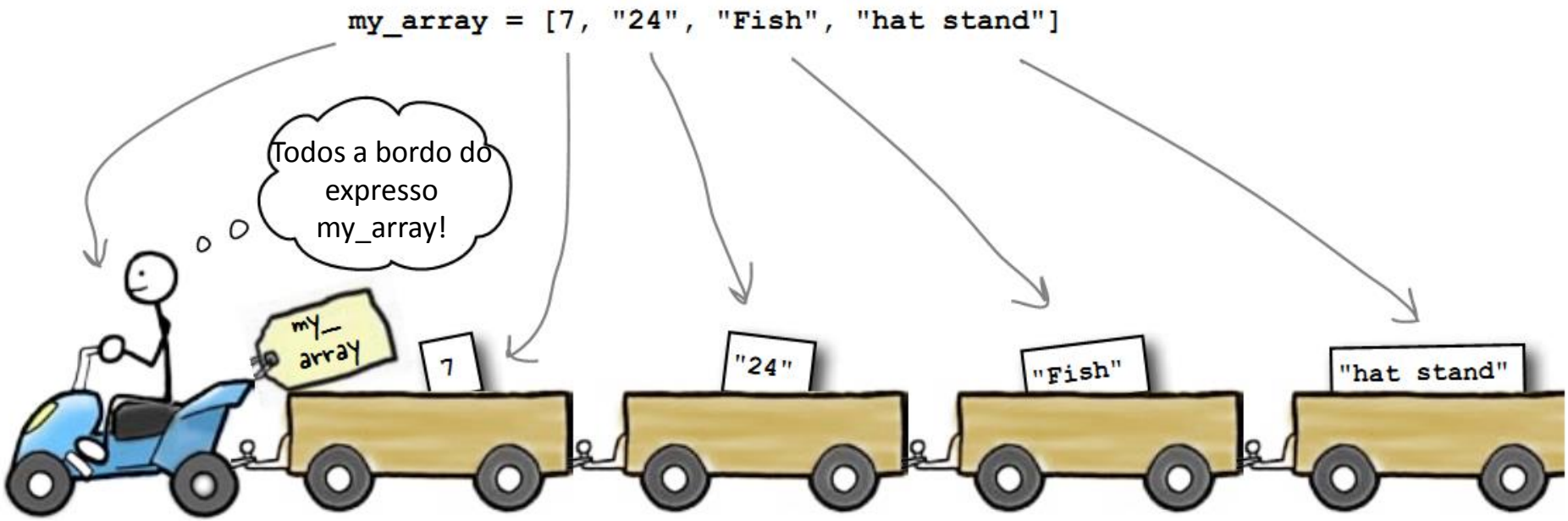
Trem de dados



Aqui vem o trem de dados

Cada vagão tem sua informação

Trem de dados



O trem de dados my_array é uma única variável

Os elementos podem ser de qualquer tipo, inclusive lista

my_array = [1, 2, 3, [“a”, 3.2]]

Posso engatar vagões

```
my_words = ["Dudes", "and"]
```

Dê um nome ao trem de dados

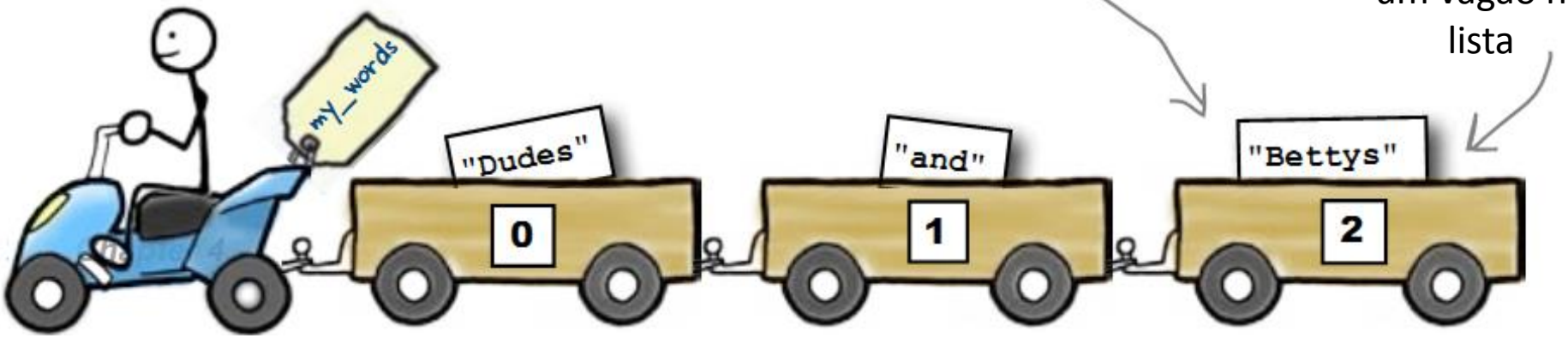
Atribua uma lista de dados


```
>>> print(my_words[0])
Dudes
>>> print(my_words[1])
and
```

Como posso acrescentar um vagão com “Bettys”?

Posso engatar vagões com append

```
>>> my_words.append("Bettys")
>>> print(my_words[2])
Bettys
```



Listas

- Uma lista vazia

```
>>> lista = []
```

- Uma lista com três notas

```
>>> notas = [7.5, 9, 8.3]
```

- Acessando uma nota

```
>>> print (notas[0])
7.5
```

- Mudando a primeira nota

```
>>> notas [0] = 8.7
>>> print (notas[0])
8.7
```

Listas

- Índice primeiro e último elemento
  - Primeiro elemento: **0**
  - Último elemento: **-1**
- “+” pode ser usado para concatenar
  - >> **lista = lista + [1]*3**
- “del” remove um elemento da lista
  - **del lista[1]**
- Notação de fatiamento
  - **lista[2:]**
  - **lista[::-1]**
- Atribuição com fatiamento
  - **lista[:2] = “xyz”**
  - **lista[1:3] = [2,3]**
- Analisar se um elemento pertence a lista
  - **2 in lista**
  - **“a” in “abcd”**
  - **7 in [3, 5, 8, “a”]**

Listas

- Funções min(), max(), len()
  - **min(lista), max(lista, len(lista)**
- Função list(), join()
  - **lista = list(string)**
  - **string = “”.join(lista)**
- Função range() – Progressão Aritmética
  - **lista = range(2, 5)**
  - **lista = range(1, 5, 2)**
  - **lista = range(10, 3, -2)**
- Função sort() e reverse() – Ordenação e Inversão
  - **lista = lista.reverse()**
  - **lista = lista.sort()**
- Outras funções:
  - **lista.extend(lista)**
  - **lista.count(elemento)**
  - **lista.index(elemento)**
  - **lista.insert(index,value)**
  - **lista.pop(index)**
  - **lista.remove(index)**
- Verifique para que serve cada uma das “Outras funções”. **Let’s go!**

Listas

- Calcule a média de 5 notas

```
notas = [6, 7, 5, 8, 9]
soma = 0
x = 0
while x < 5:
    soma += notas[x]
    x += 1
print ("Média: %5.2f" %(soma/x))
```

Obs.: x += 1 é o mesmo que x = x + 1

Listas

- Faça um programa que leia um vetor de 5 números inteiros e mostre o vetor

```
vetor = []
i = 1
while i <= 5:
    n = int(input("Digite um número: "))
    vetor.append(n)
    i = i + 1
print ("Vetor lido:", vetor)
```

Listas

- Faça um programa que leia um vetor de dez números reais e mostre-os na ordem inversa

```
vetor = []
i = 1
while i <= 10:
    n = float(input("Digite um número: "))
    vetor.append(n)
    i += 1
i = 9
while i >= 0:
    print (vetor[i])
    i -= 1
```

Listas

- Faça um programa que leia quatro notas, mostre as notas e a média na tela

```
notas = []
i = 1
while i <= 4:
    n = float(input("Nota: "))
    notas.append(n)
    i += 1
soma = 0
i = 0
while i <= 3:
    soma += notas[i]
    i += 1
print ("Notas:", notas)
print ("Média: %4.2f" %(soma/4))
```

Listas

- Outra forma de fazer o mesmo

```
notas = []
soma = 0
i = 1
while i <= 4:
    n = float(input("Nota: "))
    notas.append(n)
    soma += n
    i += 1
print ("Notas:", notas)
print ("Média: %4.2f" %(soma/4))
```

Algorithm Exercise

Listas

- Faça um Programa que leia um vetor de 10 caracteres minúsculos, e diga quantas consoantes foram lidas.

```
letras = []
i = 1
while i <= 10:
    letras.append(input("Letra: "))
    i += 1
i = 0
cont = 0
while i <= 9:
    if letras[i] not in 'aeiou':
        cont += 1
    i += 1
print ("Foram lidos %d consoantes" %cont)
```

Exercícios

- Escreva um programa que intercale os elementos de duas listas l1 e l2.
  - Exemplo: para l1 = [1,2,3] e I2 = ['a','b','c','d'], o programa deve computar a lista [1,'a',2,'b',3,'c','d']