

# Python - Alterar itens do dicionário

Além da representação literal do dicionário, onde colocamos pares chave: valor separados por vírgula entre chaves, podemos criar um objeto de dicionário com a função `dict()` integrada.

## Dicionário Vazio

Usar a função `dict()` sem nenhum argumento cria um objeto de dicionário vazio. É equivalente a não colocar nada entre chaves.

### Exemplo

```
d1 = dict()
d2 = {}
print ('d1: ', d1)
print ('d2: ', d2)
```

Ele produzirá a seguinte **saída** -

```
d1: {}
d2: {}
```

## Dicionário da lista de tuplas

A função `dict()` constrói um dicionário a partir de uma lista ou tupla de tuplas de dois itens. O primeiro item de uma tupla é tratado como chave e o segundo como seu valor.

### Exemplo

```
d1=dict([('a', 100), ('b', 200)])
d2 = dict (('a', 'one'), ('b', 'two'))
print ('d1: ', d1)
print ('d2: ', d2)
```

Ele produzirá a seguinte **saída** -

```
d1: {'a': 100, 'b': 200}
d2: {'a': 'one', 'b': 'two'}
```



## Dicionário de argumentos de palavras-chave

A função `dict()` pode receber qualquer número de argumentos de palavras-chave com pares `nome=valor`. Retorna um objeto de dicionário com o nome como chave e o associa ao valor.

### Exemplo

```
d1=dict(a= 100, b=200)
d2 = dict(a='one', b='two')
print ('d1: ', d1)
print ('d2: ', d2)
```

Ele produzirá a seguinte **saída** -

```
d1: {'a': 100, 'b': 200}
d2: {'a': 'one', 'b': 'two'}
```