

this is the item going to added to a list

```
l = [i for i in range(1,11)]  
print(l)
```

The image shows a Python list comprehension: `l = [i for i in range(1,11)]`. The variable `i` inside the square brackets is enclosed in a green box. A red arrow points from this box to a handwritten red note above the code that says "this is the item going to added to a list". The entire list comprehension expression `[i for i in range(1,11)]` is enclosed in a green oval. A blue arrow points from the `i` inside the brackets to the `i` in the `range` function.

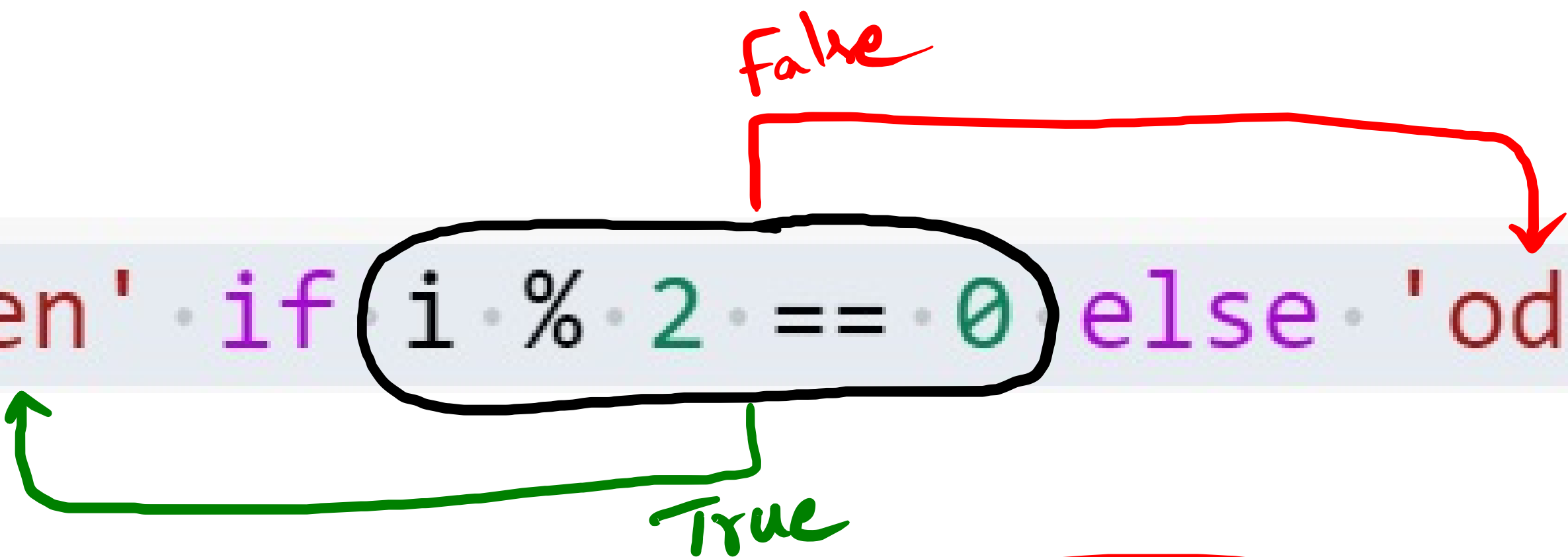
```
l = [i for i in range(1,11) if i % 2 == 0]
```

The diagram illustrates the execution of the list comprehension `l = [i for i in range(1,11) if i % 2 == 0]`. The variable `i` is circled in blue. A red arrow points from the `if` condition to the word `False`, indicating that values where the condition is false are not included in the list. A green arrow points from the `if` condition to the word `True`, indicating that values where the condition is true are included. A black oval highlights the condition `i % 2 == 0`. A black arrow points from the `if` keyword to the start of the list, and a green arrow points from the `True` label to the same start, showing the flow of elements that satisfy the condition into the list.

`'even' if i % 2 == 0 else 'odd'`

True

False

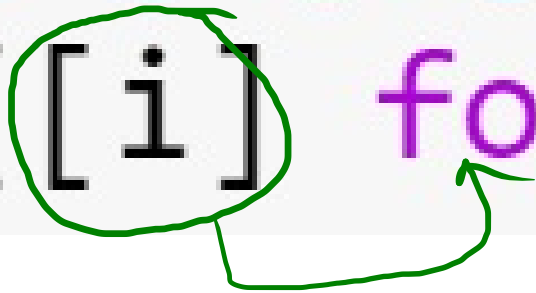


short form of if - else

```
l=[ 'even' if i % 2 == 0 else 'odd' for i in range(1,11) ]
```

→ for each iteration this portion of the code will be executed

```
l = [[i] for i in range(1,11)]
```

A green circle is drawn around the inner list `[i]` in the code. An arrow points from the circle to the variable `i` in the `for` loop, highlighting that the list contains the current value of `i`.

```
l=[ [j for j in range(i)] for i in range(1,5)]
```

↓
for each iteration this portion code result will
be added to a list.

```
l=[parent for parent in range(1,6) for child in range(parent)]
```



```
l=[[i,j,k] for i in range(2) for j in range(2) for k in range(2)]
```



The diagram shows the list comprehension `l=[[i,j,k] for i in range(2) for j in range(2) for k in range(2)]`. The expression `[i,j,k]` is circled in black. The entire comprehension is circled in red. The `for i in range(2)` loop is circled in blue, the `for j in range(2)` loop is circled in green, and the `for k in range(2)` loop is circled in purple. A black arrow points from the `[i,j,k]` list to the `for k in range(2)` loop.

```
l = []  
for i in range(2):  
    for j in range(2):  
        for k in range(2):  
            l.append([i,j,k])
```