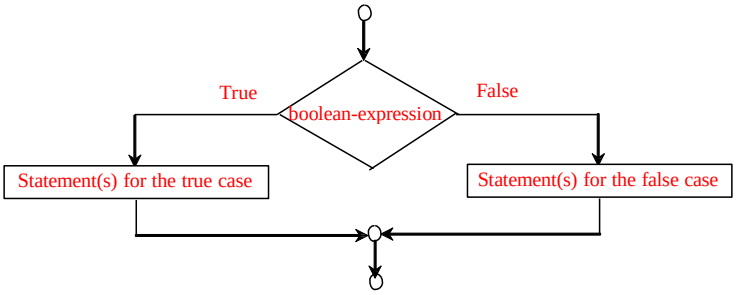
Handout 3

Selection and conditional statements

The **`if`** statement is used for conditional execution.

if boolean-expression:
  statement(s)-for-the-true-case
else:
  statement(s)-for-the-false-case



Examples: notice the indentation:

1.

```
if i > 0:
print("i is positive")
```

(a) Wrong

```
if i > 0:
    print("i is positive")
```

(b) Correct

2.

```
if radius >= 0:
    area = radius * radius * math.pi
    print("The area of a circle with radius", radius, "is", area)
else:
    print("Negative input")
```

3.

```
if time < limit:
      print ("You made it.")
      if (limit – time >= 10 ):
            bonus = 100;
      else:
            bonus = 50;
else:
      print("You missed the deadline.");
      bonus = 0;
```

TRUE AND FALSE IN PYTHON

In the context of Boolean operations, and also when expressions are used by control flow statements,
The following values are interpreted as **`False`**:

- **`False`, `None`**, numeric **zero** of all types, and **empty** strings and **empty** containers (including strings, tuples, lists, dictionaries, sets).

All values not interpreted as **`False`**, are interpreted as **`True`**

GENERAL FORM

The general form is specified in documentation as follows

```
if_stmt ::=  "if" expression ":" suite
             ("elif" expression ":" suite)*
             ["else" ":" suite]
```

Here, * means repeated 0 or more times
[] means - optional

It selects exactly one of the suites by evaluating the expressions one by one until one is found to be true (see section Boolean operations for the definition of true and false); then that suite is executed (and no other part of the if statement is executed or evaluated). If all expressions are false, the suite of the else clause, if present, is executed.

CONDITIONAL OPERATOR

Syntax: `expression1 if boolean-expression else expression2`
Shorthand for:

```
if boolean-expression:
      expression1
else:
      expression2
```

Example:

The three statements below produce identical computation

```
1. if num % 2 == 0:
      print(str(num) + "is even")
   else:
      print(str(num) + "is odd")

2. print(str(num) + " is even")  if (num % 2 == 0) else print
   (str(num) +  " is odd")

3. print(str(num) + " is even" if (num % 2 == 0) else str(num) +
   " is odd")
```

BOOLEAN OPERATIONS

In the **order of precedence** in evaluation, assuming a, b are expressions:

`not a`	Negation of a
`a and b`	`True` if both a and b are True-equivalent, and False otherwise
`a or b`	`True` if at least one of a or b is True-equivalent, and False otherwise

PRACTICE PROBLEMS ON CONDITIONAL STATEMENTS

1. Assume variables `name`, `distance`, `rating and waitTime` refer to the name, distance in miles, 5-based integer rating of a restaurant and the expected time one must wait before getting a table:
   Write a conditional that prints 'let's go to ' and the name of the restaurant if
   - if a restaurant is within a 2-mile distance, or
   - if it is between 5 and 10 miles away and has a rating no less than 4, or
   - if it has a 5 star rating and the wait time is under 20 minutes, but it is not farther than 1 mile away.

PRACTICE PROBLEMS ON STRINGS AND CONDITIONALS

2. Write a program that will display a number corresponding to a letter on a telephone number pad, shown in the figure.
   The program interaction is demonstrated below:


```
enter a character s
s is 7 on the keypad
```

   Avoid writing a long conditional.
   My solution takes 7 lines of code, including input and output statements.
   Hint: *use the ASCII code of the character to map it into a number.*

3. Given a keyword and some text, find the keyword and the two words that surround it in the sentence. Display the word, the surrounding words and … for the rest of the text. If the keyword is the first or the last word in the sentence, there will be only one surrounding word. If the keyword does not appear in the sentence,

For example, given the following text (entered without line breaks)

```
Python is an experiment in how much freedom programmers
need. Too much freedom and nobody can read another's
code; too little and expressiveness is endangered. -
Guido van Rossum
```

And keyword `freedom`, the program should output

```
. . . much FREEDOM programmers . . .
```

for keyword `python,` the output should be

```
PYTHON is ...
```

`Make sure to make the program case-insensitive` (i.e. treat python and Python as the same) and find the whole word and not a part of it (i.e., when looking for keyword `I,` don't find it in the word `is).`