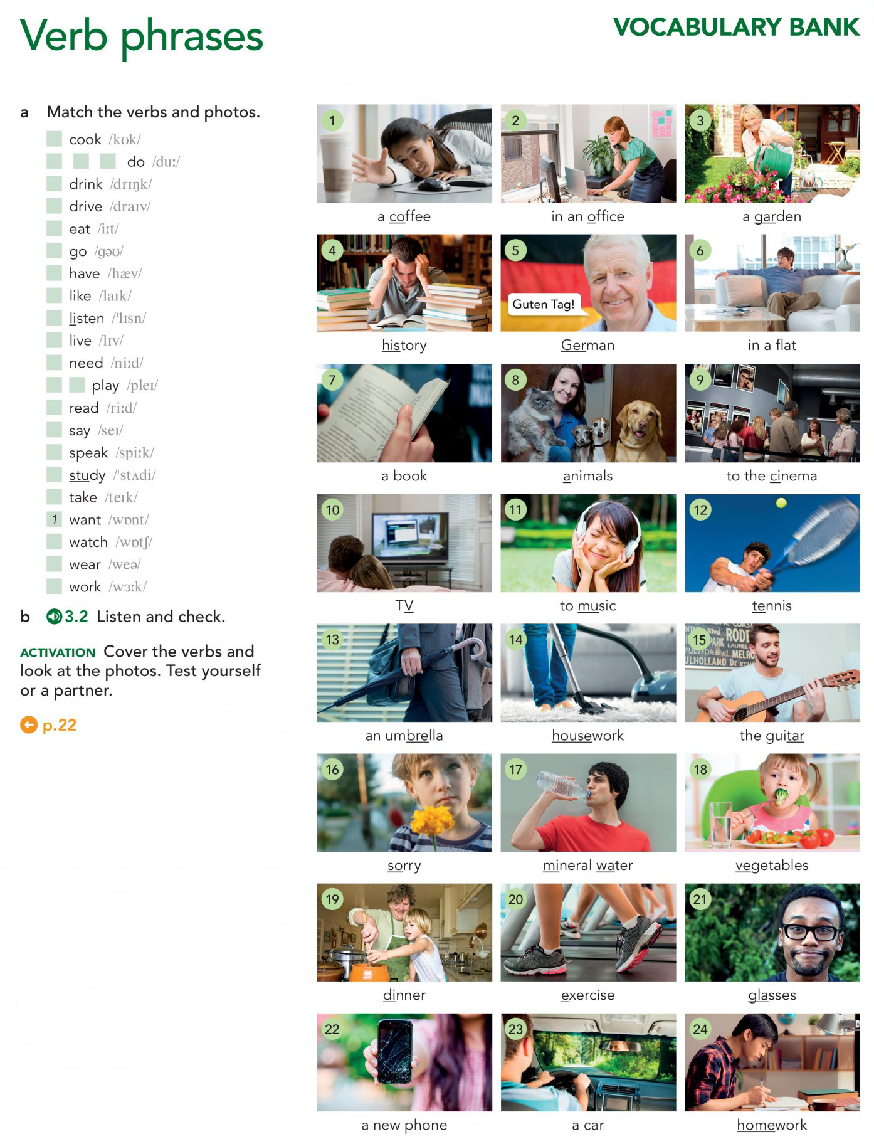


**Welcome to English Programming**

**Week 4 Class 1**

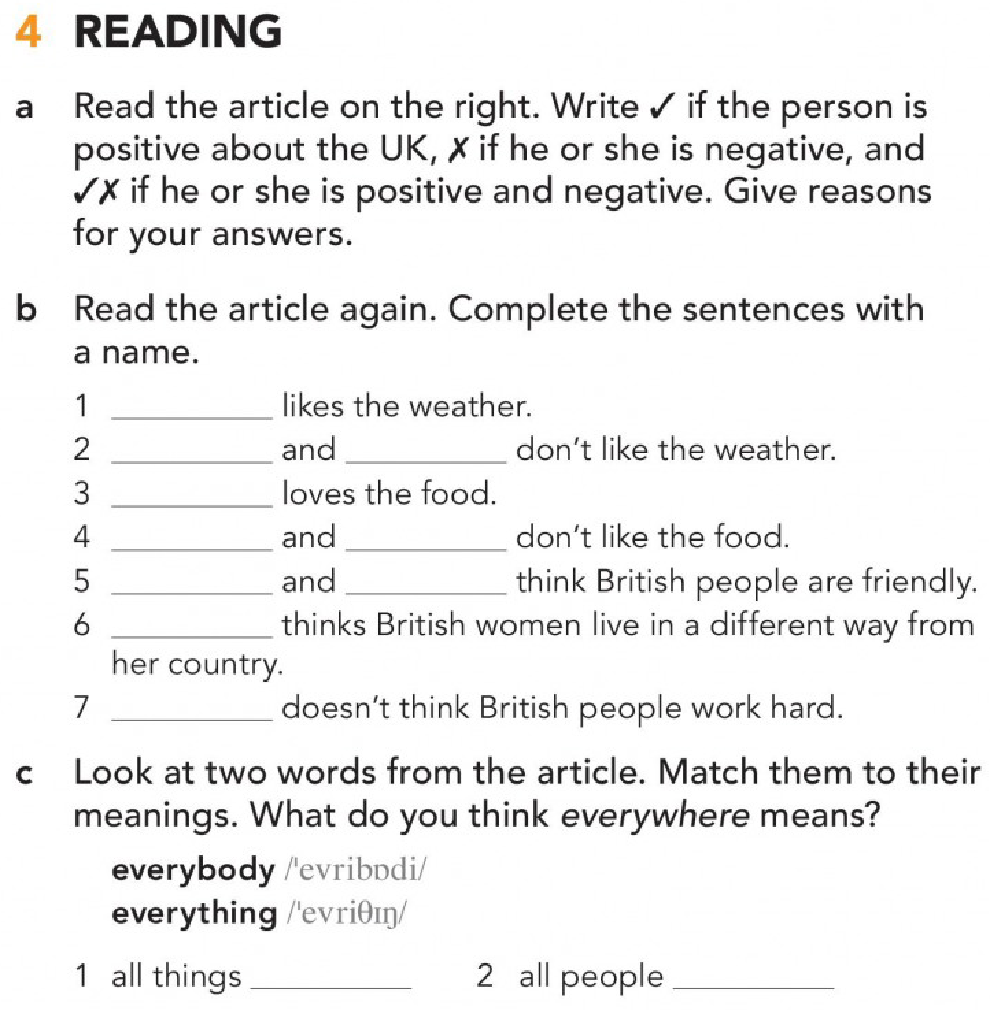
**Student Worksheet**

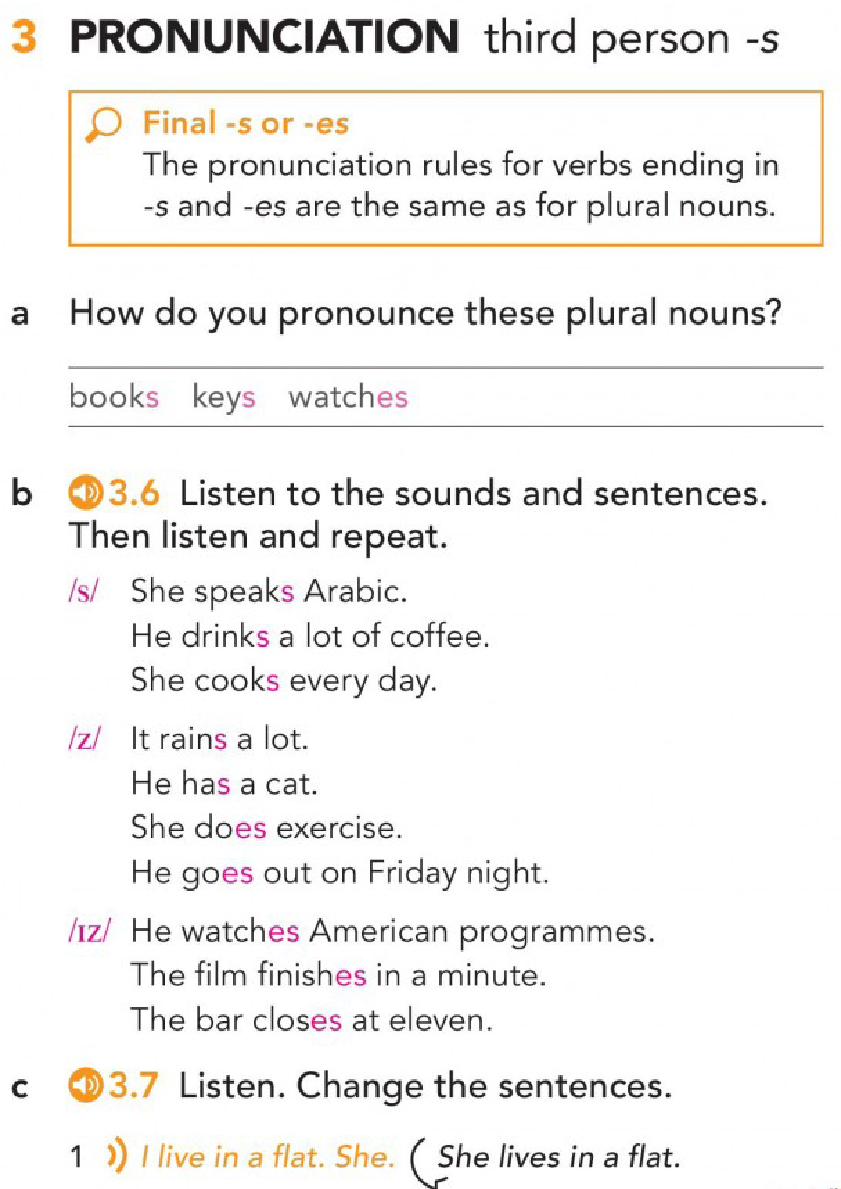
Materials for the class

**Part 1. Language Skills**

Graphical user interface, application, Word

Description automatically generated

****

****

**Text

Description automatically generated**

**Practice**

****

**More Practice**

****

**Part 2. Reading**

**BEFORE READING**

**Vocabulary**

You will read about STRINGS, and the following words (wordcloud) appear in the text. Understanding those words will help understand better the text, so complete the exercises below.

|  |
| --- |
| Graphical user interface, text  Description automatically generated |

Parts of speech.

The words in blue are \_\_\_\_\_\_\_\_\_\_\_\_\_.

The words in red are \_\_\_\_\_\_\_\_\_\_\_\_\_.

The words in pink are \_\_\_\_\_\_\_\_\_\_\_\_\_.

Which of those words are cognates?

|  |  |  |
| --- | --- | --- |
| **COGNATES** | | |
|  |  |  |

Complete the chart to form word families.

|  |  |  |  |
| --- | --- | --- | --- |
| **verb** | **noun** | **adjective** | **adverb** |
| \_\_\_\_\_\_\_\_\_\_\_\_\_ | replacement | \_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_ | ------ |
| define | \_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_ | ------ |
| \_\_\_\_\_\_\_\_\_\_\_\_\_ | Specification | \_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_ |
| \_\_\_\_\_\_\_\_\_\_\_\_\_ | value | \_\_\_\_\_\_\_\_\_\_\_\_\_ | ------ |

**Scanning**

Complete the following ideas with words taking from the text below.

string.hexdigits, string.octdigits, and string.punctuation are constants in the module about strings.

With the Formatter class you can create/customize your personal string formatting behavior.

Replacement fields are part of format strings.

**WHILE READING**

Now, it is time to read about STRINGS. Read the text and answer the questions. You can find the text in the following link: [string — Common string operations — Python 3.9.4 documentation](https://docs.python.org/3/library/string.html)

**string** — Common string operations

1. **String constants.** The constants defined in this module are:
   1. string.ascii\_letters
   2. string.ascii\_lowercase
   3. string.ascii\_uppercase
   4. string.digits
   5. string.hexdigits
   6. string.octdigits
   7. string.punctuation
   8. string.printable
   9. string.whitespace
2. **Custom String Formatting.** 
   1. The built-in string class provides the ability to do complex variable substitutions and value formatting via the format() method described in PEP 3101.
   2. The Formatter class in the string module allows you to create and customize your own string formatting behaviors using the same implementation as the built-in format() method.
   3. class string.Formatter The Formatter class has the 8 public methods.
3. **Format String Syntax.**

(Paragraph 1) The str.format() method and the Formatter class share the same syntax for format strings (although in the case of Formatter, subclasses can define their own format string syntax). The syntax is related to that of formatted string literals, but it is less sophisticated and, in particular, does not support arbitrary expressions.

(Paragraph 2) Format strings contain “replacement fields” surrounded by curly braces {}. Anything that is not contained in braces is considered literal text, which is copied unchanged to the output. If you need to include a brace character in the literal text, it can be escaped by doubling: {{ and }}.

(Paragraph 3) The grammar for a replacement field is as follows:

Graphical user interface, application, Word

Description automatically generated

(Paragraph 4) In less formal terms, the replacement field can start with a field\_name that specifies the object whose value is to be formatted and inserted into the output instead of the replacement field. The field\_name is optionally followed by a conversion field, which is preceded by an exclamation point '!', and a format\_spec, which is preceded by a colon ':'. These specify a non-default format for the replacement value.

(Paragraph 5) The field\_name itself begins with an arg\_name that is either a number or a keyword. If it’s a number, it refers to a positional argument, and if it’s a keyword, it refers to a named keyword argument. If the numerical arg\_names in a format string are 0, 1, 2, … in sequence, they can all be omitted (not just some) and the numbers 0, 1, 2, … will be automatically inserted in that order. Because arg\_name is not quote-delimited, it is not possible to specify arbitrary dictionary keys (e.g., the strings '10' or ':-]') within a format string. The arg\_name can be followed by any number of index or attribute expressions. An expression of the form '.name' selects the named attribute using getattr(), while an expression of the form '[index]' does an index lookup using \_\_getitem\_\_().

(Paragraph 6) The conversion field causes a type coercion before formatting. Normally, the job of formatting a value is done by the \_\_format\_\_() method of the value itself. However, in some cases it is desirable to force a type to be formatted as a string, overriding its own definition of formatting. By converting the value to a string before calling \_\_format\_\_(), the normal formatting logic is bypassed.

(Paragraph 7) Three conversion flags are currently supported: '!s' which calls str() on the value, '!r' which calls repr() and '!a' which calls ascii().

(Paragraph 8) The format\_spec field contains a specification of how the value should be presented, including such details as field width, alignment, padding, decimal precision and so on. Each value type can define its own “formatting mini-language” or interpretation of the format\_spec.

(Paragraph 9) Most built-in types support a common formatting mini-language, which is described in the next section.

(Paragraph 10) A format\_spec field can also include nested replacement fields within it. These nested replacement fields may contain a field name, conversion flag and format specification, but deeper nesting is not allowed. The replacement fields within the format\_spec are substituted before the format\_spec string is interpreted. This allows the formatting of a value to be dynamically specified.

1. **Format Specification Mini-Language**

“Format specifications” are used within replacement fields contained within a format string to define how individual values are presented (see Format String Syntax and Formatted string literals). They can also be passed directly to the built-in format() function. Each formattable type may define how the format specification is to be interpreted.

* + - 1. **Meaning from context**

Match the words with their corresponding meaning.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **WORD** |  |  | **DEFINITION** |
| a. | String (n) \_\_\_\_\_ |  | 1. | (n) the distance across something from one side to the other; extension |
| b. | Customize (v) \_\_\_\_\_  (PART 2) |  | 2. | (n) The space between the inside edge of a child element and its content. |
| c. | Behavior (n) \_\_\_\_\_  (PART 2) |  | 3. | (n) A value that a function or a method uses to perform operations or calculations. It is specific to the function or method. They include numbers, text, cell references, and names. |
| d. | Syntax (n) \_\_\_\_\_  (PART 3) |  | 4. | (adj) embedded |
| e. | Argument (n) \_\_\_\_\_  (PART 3, paragraph 5) |  | 5. | (v) personalize |
| f. | Width (n) \_\_\_\_\_  (PART 3, paragraph 8) |  | 6. | (n) a particular way of acting |
| g. | Padding \_\_\_\_\_  (PART 3, paragraph 8) |  | 7. | (n) A group of characters or character bytes handled as a single entity. |
| h. | Nested (adj) \_\_\_\_\_  (PART 3, paragraph 10) |  | 8. | (n) The rules governing the formation of a command-line statement, including the order in which a command must be typed, and the elements that follow the command. |

Based on the information from the text, answer the following questions.

What are ‘braces’? \_\_\_\_\_

What is ‘exclamation point’? \_\_\_\_\_

What is ‘colon’? \_\_\_\_\_

* + - 1. **Reading comprehension**

Based on the information from the text, decide if the following sentences are true (T) or false (F).

When you copy to the output anything considered literal text, this does not change. T/F

To format and insert the value of an object into the output you use the replacement field only. T/F

A field name, conversion flag, and format specification can be found in replacement fields, and you can embed more information. T/F

You use ‘format specifications’ to determine the type of presentation of individual values. T/F

**Part 2. Reading**

**Answer Key**

**BEFORE READING**

**Vocabulary**

Parts of speech.

The words in blue are verbs.

The words in red are nouns.

The words in pink are adjectives.

Which of those words are cognates?

|  |  |
| --- | --- |
| **COGNATES** | |
| Conversion  Type  Replacement  Specification | Format  Method  Class  index |

Complete the chart to form word families.

|  |  |  |  |
| --- | --- | --- | --- |
| **verb** | **noun** | **adjective** | **adverb** |
| replace | replacement | replaceable  replaced | ------ |
| define | definition | Defined | ------ |
| specify | Specification | Specific  specified | specifically |
| value | value | valuable | ------ |

**Scanning**

Complete the following ideas with words taking from the text below.

string.hexdigits, string.octdigits, and string.punctuation are constants in the module about strings.

With the Formatter class you can create/customize your personal string formatting behavior.

Replacement fields are part of format strings.

**WHILE READING**

* + - 1. **Meaning from context**

Match the words with their corresponding meaning.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **WORD** |  |  | **DEFINITION** |
| a. | String (n) 7 |  | 1. | (n) the distance across something from one side to the other; extension |
| b. | Customize (v) 5  (PART 2) |  | 2. | (n) The space between the inside edge of a child element and its content. |
| c. | Behavior (n) 6  (PART 2) |  | 3. | (n) A value that a function or a method uses to perform operations or calculations. It is specific to the function or method. They include numbers, text, cell references, and names. |
| d. | Syntax (n) 8  (PART 3) |  | 4. | (adj) embedded |
| e. | Argument (n) 3  (PART 3, paragraph 5) |  | 5. | (v) personalize |
| f. | Width (n) 1  (PART 3, paragraph 8) |  | 6. | (n) a particular way of acting |
| g. | Padding 2  (PART 3, paragraph 8) |  | 7. | (n) A group of characters or character bytes handled as a single entity. |
| h. | Nested (adj) 4  (PART 3, paragraph 10) |  | 8. | (n) The rules governing the formation of a command-line statement, including the order in which a command must be typed, and the elements that follow the command. |

Based on the information from the text, answer the following questions.

What are ‘braces’? {}

What is ‘exclamation point’? !

What is ‘colon’? :

* + - 1. **Reading comprehension**

Based on the information from the text, decide if the following sentences are true (T) or false (F).

When you copy to the output anything considered literal text, this does not change. T/F

To format and insert the value of an object into the output you use the replacement field only. T/F

A field name, conversion flag, and format specification can be found in replacement fields, and you can embed more information. T/F

You use ‘format specifications’ to determine the type of presentation of individual values. T/F