**Python**

**Identifiers:** Identifiers in Python are names used to identify a variable, function, class, module, or other objects. They help in referencing those objects throughout the code.

Some of the rules of picking identifiers are:

* Must begin with a letter (A-Z or a-z) or an underscore (\_).
* Cannot start with a digit (0-9).
* Can contain letters, digits, and underscores after the first character.
* Cannot contain special characters like !, @, #, $, %, etc.
* Identifiers are case-sensitive (e.g., myVar, MyVar, and MYVAR are different).
* Cannot be a reserved keyword (e.g., class, def, if, else, etc.).
* Avoid using single-character identifiers, except for loop variables (e.g., i, j).

We have some kind of variables that start with underscore as:

* \_a = private variable
* \_\_a= protected/ strongly private variable
* \_\_a\_\_= magic/ dunder variable

**Keywords:**

Keywords in Python are reserved words that have a specific meaning and purpose in the language. These words are used to define the structure and control flow of Python programs. They form the basic building blocks of Python's syntax and cannot be used as identifiers for variables, functions, or classes. Some of the keywords of python are:

False, None, True, and, as, assert, async, await, break, class, continue, def, del, elif, else, except, finally, for, from, global, if, import, in, is, lambda, nonlocal, not, or, pass, raise, return, try, while, with, yield.

**Data types:**

Python has several built-in **data types**, which represent different kinds of values that can be stored and manipulated in a program. Here's an overview of Python's most common data types:

**Fundamental/ main data types**: (Int, float, complex, boot, str )

**Other datatypes**: byte, bytearray, range, list, tuple, set, frozenset, dict, none

**Int**: Basically used to denote integer/whole number data type. Eg: 5 is integer but 5.0 is not an integer.

* It can be written in binary, decimal, octal or hexadecimal.
* By default it is in decimal.
* To write the character in binary we use ‘0b’,octal ‘0o’, hexa ‘0x’ and by default it is in decimal (it is not case sensitive).
* To convert the output of a= 1234 in other format we do “print(hex(a))” for changing the output in hexadecimal, “print(oct(a))” for octal, “print(bin(a))” for binary and “print(a)” for decimal.

**Float:** The integer number that have some numbers after the point or decimal then it is said to be float number.

**HTML**

**Text Formatting Elements**

**<b> (Bold)**: This tag makes text bold for styling purposes but without indicating any importance or extra meaning.

**<strong> (Strong Importance)**: This tag makes text bold **and** indicates that the content is of strong importance or seriousness. Search engines and screen readers might give it more emphasis.

**Difference**: While both tags make the text bold, **<strong>** conveys importance, while **<b>** is purely for visual formatting.

**<i> (Italics)**: This tag italicizes text for styling, without implying any emphasis or stress.

**<em> (Emphasized Text)**: This tag italicizes text **and** adds emphasis, suggesting the text should be stressed or given more importance in reading.

**Difference**: Both tags display text in italics, but <em> indicates that the content is important or should be emphasized.

**<sup> (Superscript)**: This tag raises text to appear slightly above the baseline. It is often used for footnotes or mathematical exponents.

**<sub> (Subscript)**: This tag lowers text to appear slightly below the baseline. It is typically used for chemical formulas or in mathematical contexts.

**<blockquote> (Block Quote)**: This tag is used for longer quotes, typically displayed as a block element, often indented. It is used for quoting entire sections or paragraphs of text.

**<q> (Inline Quote)**: This tag is used for short, inline quotations. Quotation marks are added automatically by the browser.

**Difference**: <blockquote> is for block-level (long) quotations, whereas <q> is for short, inline quotations.

**Links and URLs**

**The anchor tag <a>:** The HTML <a> tag (anchor tag) is used to create hyperlinks that allow users to navigate to different pages, sections, files, or even to send emails. Here’s how you can use the <a> tag for different purposes:

**1. Linking to External Pages**

We can create a hyperlink to an external website using the <a> tag with the href attribute, which specifies the URL of the page we're linking to. Eg: <a href="https://www.example.com">Visit Example Website</a>

When the link is clicked, the user is taken to "<https://www.example.com>".

**2. Linking to Internal Sections (Anchors)**

We can link to a specific section within the same page using the id attribute of the target element and referring to it with the # symbol in the href attribute

**3. Linking to Files (Documents, PDFs, Images)**

We can use the <a> tag to link to files like PDFs, documents, or images. The href attribute should point to the file’s location. Eg: <a href="files/resume.pdf" download>Download My Resume</a>

Clicking this link will download the file "resume.pdf". You can also omit download if you just want to open the file in the browser.

**4. Email Links (mailto:)**

To create a link that opens the user's email client to send an email, you use the mailto: protocol in the href attribute. Eg: <a href="mailto:example@example.com">Send me an email</a>

Clicking this link will open the default email client, with the recipient’s email address already filled in.







