**JSON**

*Бібліотека* ***json*** *дозволяє отримати доступ до API інформації, отримати дані у форматі, з яким можливо та зручно працювати використовуючи* ***python.***

json.dumps(obj) – object of any type to string using the conversion table

json.loads(obj) – string to object of the appropriate type using the conversion table

json.dump(obj, file) – converts object to string and writes to file

json.load( file) - reads file, returns file content in form of object converted to appropriate type

**HTML**

*Дозволяє здійснювати парсинг html та xhtml файлів*

class html.parser.HTMLParser(contained in html) – creates parser, that is able to loads html files and evaluate start tags, end tags, data, comments and other markups

HTMLParser.feed(text) – loads data to the parser, data must be str type

HTMLParser.reset() – reset the instance and loses all unprocessed data

HTMLParser.get\_starttag\_text() – return the text between start-tags

HTMLParser.handle\_starttag(tag, attrs) - handle the start-tag and list with tuples of its attributes in form (name, value)

HTMLParser.handle\_endtag(tag) – called to handle end-tags

HTMLParser.handle\_data(data) – called to process content of html-file

HTMLParser.handle\_comment(data) – called to process comments

**XML**

*Бібліотека* ***xml*** *дозволяє отримати доступ до API інформації представленої у форматі XML, завантажити дані у ієрархічному вигляді, з яким можливо та зручно працювати використовуючи* ***python*** *та* ***xml.***

xml.etree.ElementTree – xml represents as tree as it is hierarchical data format. Here we have two classes – ElementTree(represents whole xml tree) and Element(represents only a single node of a tree)

xml.etree.ElementTree.getroot() – gets root of xml file

xml.etree.ElementTree.getroot() [index1][index2].text – return the text between tag with index2 in root child with index1

class ElementTree:

\_setroot(element) – set the root for this xml tree

find(match, namespaces=None) – starting from the root of the tree, finds the first subelement matching match

findtext(match, default=None, namespaces=None) – returns text from the first subelement matching match

getroot() – returns the root element for this tree

parse(source, parser=None) – load an xml file to the element tree, source may be file name or file object

class Element (tag, attrib={}, \*\*extra):

tag – a string, that mean what kind of data represents

attrib – a dictionary, contains element attributes

-----------------------------------------------------------------------------

clear() – clears the element

get(key, default=None) – gets the element attribute with name key, returns the attribute value

items() - returns the elements attribute names as a list

find(match, namespaces=None) – the same as in ElementTree

findtext(match, namespaces=None) – the same as in ElementTree