

Logging

למה צריך?

- אנחנו רוצים לעקוב ולתעד ריצת התוכנה (למה?)
- רוצים להיות ערים לבעיות פוטנציאליות או קיימות
- אנחנו לא רוצים:
 - להדפיס ללקוח כל מה שקורה שם
 - לעצור את התוכנה על כל דיווח
 - להיות תלויים באובייקטים של exception

הודעות שונות (לפי דרגות)

- `print(" 😊 ")` - תוצאה סופית
- `logging.debug("")` - הכל טוב, דיווח שוטף מפורט
- `logging.info("")` - הכל טוב, דיווח שוטף בקטנה
- `logging.warning("")` - משהו לא עובד בדיוק כמו שציפינו, לשים לב
- `logging.error("")` - משהו גורם לפונקציה שלא תעבוד
- `logging.exception("")` - אותו דבר רק קשור לאקספצן מסוים
- `logging.critical("")` - יש בעיה שתגרום לכל התוכנה לא לעבוד
- `raise exception` - אי אפשר להמשיך ככה!

נכתוב הודעות

```
import logging
```

צריך תמיד לייבא את הסיפריה. נמציא פונקציה פשוטה כזו

```
def add_one(x):  
    return x+1
```

נדפיס אותו בתור אזהרה.

```
logging.warning(add_one(6))
```

אבל לא באמת מתאים פה להזהיר

```
logging.debug(add_one(6))
```

מה קרה?

דרגות שונות

- יש דרגות שונות של הודעות וברירת המחדל של המערכת להדפיס רק הודעות שהן גבוהות יותר מ-warning

- נאתחל ונחזור שנית על הכל מה-import ומיד אחרי היבוא נגדיר

```
logging.basicConfig(level = logging.DEBUG)
```

- וננסה שוב

```
logging.debug(6+1)
```

דוגמה לכתיבת לוגים על סדרת פעולות

```
import logging
logging.basicConfig(level = logging.DEBUG)
def add_evens(nums):
    _sum = 0
    for i in nums:
        if i%2:
            logging.warning(str(i) + " is uneven. will not be included in sum!")
        else:
            logging.debug("adding " + str(i) + " to sum, because it is even")
            _sum += i
            logging.info(str(i) + " added")
    print(_sum)
```

עכשיו אפשר לשחק עם דרגות ועוד דברים. למשל...

כתיבת לוגים לקובץ

```
logging.basicConfig(level = logging.INFO, filename = "add_evens.log")
```

- מה מודפס עכשיו בקונסול?

- נפתח את הקובץ ונבדוק מה הודפס שם

```
WARNING:root:1 is uneven. will not be included in sum!
```

```
WARNING:root:3 is uneven. will not be included in sum!
```

```
INFO:root:0 added
```

```
WARNING:root:1 is uneven. will not be included in sum!
```

```
INFO:root:2 added
```

```
WARNING:root:3 is uneven. will not be included in sum!
```

- טוב מאוד. אבל לא שיא היופי. היינו רוצים לשלוט יותר בנוסח הלוג

פורמטים

ממה הייתה מורכבת ההודעה שקיבלנו?

Level name	Logger name	Message
WARNING:	root:	1 is uneven. will not be included in sum!

```
logging.basicConfig(level = logging.INFO,  
                    filename = 'add_evens.log',  
                    format = "%(asctime)s:%(levelname)s:%(message)s")
```

• נריץ שוב ונבדוק את הלוג

- <https://docs.python.org/3/library/logging.html#logrecord-attributes>

loggers

- new file:

```
import logging
```

```
logging.basicConfig(filename="div.log", level = logging.INFO)
```

```
def div(x,y):  
    logging.info("dividing " + str(x) + " by " + str(y))  
    if y==0:  
        logging.critical("dividing by zero!!!")  
        return x  
    return x/y
```

נוסיף אותו לקוד הקודם שלנו, מה הלוג?

```
import logging
import div as d

logging.basicConfig(level = logging.INFO, filename = 'add_evens.log', format = "%(asctime)s: %(levelname)s: %(message)s")

def add_evens(nums):
    _sum = 0
    _divided = 1
    for i in nums:
        _divided = d.div(_divided,i)
        if i%2:
            logging.warning(str(i) + " is uneven. will not be included in sum!")
        else:
            logging.debug("adding " + str(i) + " to sum, because it is even")
            _sum += i
            logging.info(str(i) + " added")
    print("result of sum is " + str(_sum))
    print("result of divided is " + str(_divided))
```

אנחנו צריכים לוגרים נפרדים!

```
div_logger = logging.getLogger(__name__)
```

```
replace logging -> div_logger
```

אבל עדיין לא פתרנו את הבעיה...

Handlers

```
file_handler = logging.fileHandler('div.log')
```

```
div_logger.addHandler(file_handler)
```

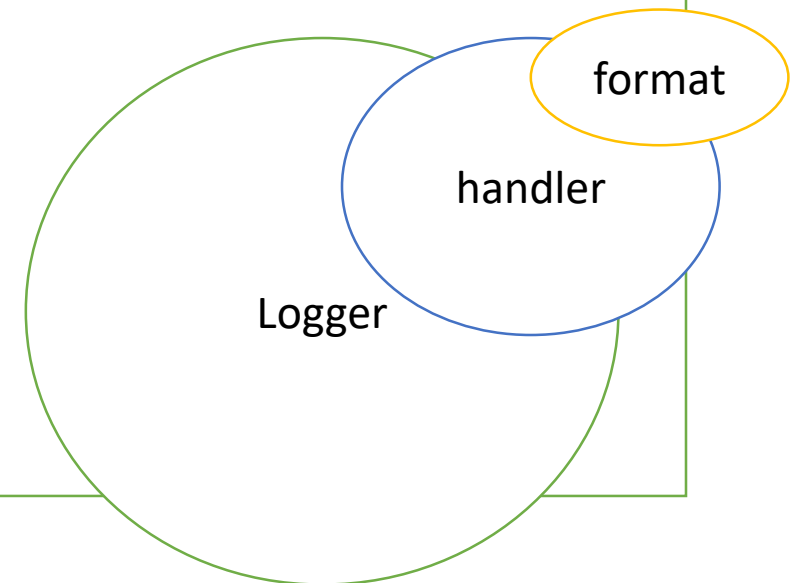
אם נרצה להוסיף פורמט ללוג שלנו

```
file_handler = logging.fileHandler('div.log')
```

```
formatter = logging.Formatter("%(asctime)s....")
```

```
file_handler.setFormatter(formatter)
```

```
div_logger.addHandler(file_handler)
```



בשביל מה כל זה?

- האם זה עזר?
- המטרה: ביטול ההורשה של `basicConfig`
- תרגיל: בואו נקנפג גם את `add_evens`

הפרדה בין הודעות לקונסול וללוג בקובץ

```
div_logger.setLevel(logging.DEBUG)
```

```
file_handler.setLevel(logging.ERROR)
```

- או להיפך (מתי נרצה כך ומתי נרצה כך?)
- האם התקבלה התוצאה הרצויה לנו? למה?
- תרגיל כיתה: נבנה את זה נכון

כמה סוגי לוגרים

```
stream_handler = logging.StreamHandler()  
logger.addHandler(stream_handler)
```

• תרגיל: log exception כנגד error log