

Date: 20th April Time: 09:30 - 11:00 & 14:15 - 15:45
Venue: Rm 4210 OR Rm 4213 Expected no. of students: 20 per class
Expected Level of students: F.2 to F.4 Context: Coding/Programming in Python
Foci: I/O, Variables, Decision Making, Functions, **random** package, Game Creation

Intended Learning Outcomes

By the end of the lesson, students should be able to/have:

- implement basic input/output statements in Python,
- manipulate data with variables in Python, combined with decision making,
- a very brief knowledge of the python **random** library and the usage of the **randint()** function,
- the skills to implement simple text-based operations in Python.

Basic Rundown

- Teachers use Mentimeter to grasp students' knowledge background.
- Teachers first demonstrate what the number guessing game is. Play it once with class.
- Teachers introduce general knowledge around programming, Python and Jupyter Notebook
- Teachers introduce the agenda of the lesson.

(The above takes ≤ 10 minutes)

- Teachers teach the foci one-by-one.
 - I/O and Variables (≤ 20 minutes)
 - * Teachers introduce the **print()** and **input()** functions.
 - * Students practice basic input and output.
 - * Teachers introduces what a variable is, and basic data types.
 - * Teachers introduce type-casting methods.
 - * Students practice I/O with type-casting methods.
 - Python **random** package and number guessing game (≤ 20 minutes)
 - * Teachers introduces the **random** package, particularly the function **random.randint()**.
 - * Using the Jupyter Notebook file provided, teachers guide students to implement their own number-guessing game.
 - Decision Making (≤ 15 minutes)
 - * Teachers introduce **if**-statement and conditions, including **or**, **and** (and possibly **not**) keywords.
 - * Students practice decision making with examples given.
- (If time permits, teachers introduce a basic look at loops)
- Teachers summarise the lesson, and tease what's to come, including loops, lists and the final game: Hangman.

(The above takes ≤ 5 minutes.)

Materials

- Computers at the venue to allow students to have hands-on experience in programming.
- A set of lecture notes to assist teachers in the lesson and students to follow along.
- Small props (paper boxes, for example) to visualise certain concepts.
- A Jupyter Notebook **.ipynb** file to allow students to code along in the lesson.
- (TBD) Prizes (snacks, for example) for answering questions.