

Date:	20 th April	Time:	09:30 - 11:00 & 14:15 - 15:45
Venue:	Computer Barn C	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, 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 (≤ 25 minutes)
 - * Teachers introduce the **print()** function.
 - * Teachers introduces what a variable is, and basic data types and arithmetic operations.
 - * Students practice basic output and arithmetic operations.
 - * Teachers introduce the **input()** function.
 - * Teachers introduce type-casting methods.
 - * Students practice I/O with type-casting methods.
 - Python **random** package (≤ 3 minutes)
 - * Teachers introduces the **random** package, particularly the **random.randint()** function.
 - Decision Making (≤ 25 minutes)
 - * Teachers introduce **if-elif-else** clauses and conditions, including **or**, **and** (and possibly **not**) keywords.
 - * Students practice decision making with the number guessing game logic.
 - Finishing up the game (Remaining time)
 - * Teachers helps students to finish up the game, and debug if necessary.
 - (If time *really* 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

- Access to 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.