| **DERSLER** | **KONULAR** | **KOMUTLAR** | **NOTLAR** |
| --- | --- | --- | --- |
| session1\_30\_01  -Introduction | pep8 kuralları  comment  variables  data types | print()  type() |  |
| session2\_31\_01  -Data Types | pep8 kuralları  comment  variables  data types | print()  type() |  |
| session3\_01\_02  -Type Conversation | Type Conversation  all types to str  float to int  int to float | input()  str()  int()  float() |  |
| session4\_02\_02  -Arithmetic Operators  & Print Function | Arithmetic Operations | print()  end=”\n”, end=”\n”  sep=” ” |  |
| session5\_03\_02-  Escape Sequences & Comparison Operators | Escape Sequences  hexa values  ASCII table  Comparison Operators  Boolean  Logic Expressions | modulus  float division  \n, \t, \b  round()  ord(), chr()  <, >, ==, !=, <=, >=  bool()  not, and, or, None, True, False |  |
| session6\_04\_02  -Logic Expressions & String1 | Indexing & Slicing | index()  [ ],[ : : ]  lower()  upper()  title()  capitalize()  swapcase() |  |
| session7\_16\_02  -Strings2 | if, elif, else | isalnum()  isnumeric()  isalpha()  replace()  isascii()  count()  find()  split()  strip(), lstrip(), rstrip()  startswith(), endswith() |  |
| session8\_17\_02  -Lists1 | Lists  Basic Operations With Lists | list(), [ ]  len()  append()  insert()  remove()  sort(), (reverse=True) |  |
| session9\_18\_02  -Lists2 | inserting picture, link  “{:<20}”.format()  “{:>20}”.format()  “{:^20}”.format()  center() | pop()  clear()  del()  count()  extend()  join() |  |
| session10\_21\_02  -Tuples |  | tuple(), ()  range() |  |
| session11\_23\_02  -Dictionaries | key oluşturma  value oluşturma  nested dicts | dict(), { }  { }.fromkeys()  items(), keys(), values()  update()  get()  in, not in |  |
| session12\_24\_02  -Sets | Sets  Union, Intersection, Difference | set(), { }  difference(), -  intersection(), &  union(), |  symmetric\_difference()  difference\_update()  intersection\_update()  add()  issubset()  discard() |  |
| session13\_25\_02  -while loop | Loops – while  random modülü | import  random.random()  random.seed()  random.randint()  max(), min() |  |
| session14\_27\_02  -for loop | Loos – for  break, continue | sum()  zip() |  |
| session15\_28\_02  -list comp & loops recap | List Comprehension  nested loops  Armstrong numbers | zip()  anumerate() |  |
| session16\_02\_03  \_functions1 | Functions  .\_\_doc\_\_  Positional and Keyword Arguments | def, return  multiply()  help() |  |
| session17\_03\_03  \_functions2 | Arbitrary Number of Arguments (\*args, \*\*kwargs)  Lambda  recursive function | \*\*kwargs,  kwargs.items()  filter()  map(),  lambda |  |
| session18\_04\_03  -Exception Handling & Moduls | Exception Handling  Modules  Create Modul | try, except, else, finally  import random  random.choise()  random.shuffle()  random.sample()  import string  string.punctuation  string.digits  string.ascii\_letters  pyautogui.position |  |