**Section 20: Advanced Python Objects and Data Structures**

**29.03.**

**138. Advanced Numbers**

The numbers can be converted to hex and binary format with hex(number) and bin(number)

We can use pow with a mod applied after pow(3, 4, 5) = (3\*\*4) mod 5

**139. Advanced Strings**

s.capitalize() to capitalize the first letter in a string

Remember that strings are immutable in Python

s.count(string) counts the number of occurrences of string in s

s.find(string) returns the index of the first occurrence of string in s

s.isalnum() True if all characters are alphanumeric

s.isalpha() True if all characters are alphabetic

s.islower() True if all characters are lower-cased

There are methods like: s.isspace(), s.istitle(), s.endswith(string)

s.partition(string) -> first half, string, second half -> for the first occurrence

**140. Advanced Sets**

s.clear() to remove all the elements from a set

s.copy() to create a deep copy of a set

s1.difference(s2) = s1 \ s2

s1.difference\_update(s2) will put in s1 the results of s1 \ s2

s.discard(value) removes the element from set, if isn’t there it does nothing

There also are s1.intersection(s2) and s1.intersection\_update(s2)

s1.isdisjoint(s2) True if s1 and s2 don’t have common elements

s1.issubset(s2) and s1.issuperset(s2); s1.issubset(s1) is True

There also are s1.symmetric\_difference(s2) and s1.symmetric\_difference\_update(s2)

A symmetric difference give the elements that are just in one set

s1.union(s2), but there isn’t s1.union\_update(s2), but there is s1.update(s2)

**142. Advanced Lists**

l.extend(iter) – extends list by appending elements from the iterable

l.append(iter) – appends the whole object at the end

l.insert(index, value)