1. [redacted]

```
x = 5
y = "john"
```

[redacted]

```
print(type(x))
print(type(y))
```

In this code, we first define the variables `x` and `y` as `5` and `"john"`, respectively.

We then use the `type()` function to find the datatype of each variable, and we print the results to the console using the `print()` function.

```
<class 'int'>
```

```
<class 'str'>
```

2. Valid variable names in Python can only contain letters, numbers, and underscores (_), and they cannot start with a number. Therefore, out of the given variable names:

1. `3a=10` - Invalid syntax (cannot start with a number)
2. `@abc=10` - Invalid syntax (contains an invalid character)
3. `a100=100` - Valid syntax
4. `_a984=100` - Valid syntax
5. `a9967$=100` - Invalid syntax (contains an invalid character)
6. `xyz-2=100` - Invalid syntax (contains an invalid character)

So, the valid variable names are `a100`, `_a984`, and `xyz2`.

3.

```
test_list = [1, 6, 3, 5, 3, 4]
```

[redacted]

```
if 3 in test_list:
    print("3 exists in the list")
else:
    print("3 does not exist in the list")
```

```
[redacted]
if 9 in test_list:
    print("9 exists in the list")
else:
    print("9 does not exist in the list")
```

In this code, we first define the list `test_list` as `[1, 6, 3, 5, 3, 4]`.

We then use the `in` keyword to check if the element 3 exists in the list. If it does, we print the message "3 exists in the list". Otherwise, we print the message "3 does not exist in the list".

We then repeat this process to check if the element 9 exists in the list. Since 9 is not in the list, the output of the second check will be "9 does not exist in the list".

```
3 exists in the list
```

```
9 does not exist in the list
```

4.

```
import datetime
```

```
[redacted]
name = input("Enter your name: ")
```

```
[redacted]
now = datetime.datetime.now()
```

```
[redacted]
print("Hello, " + name + "! Today is " + now.strftime("%Y-%m-%d") + ".")
```

Date = input(“Enter Date”)

Print(“Entered date is: “, Date)

5.

```
print(9//2)
print(9%2)
```

```
4
1
```

This is because `//` is the floor division operator, which divides 9 by 2 and rounds the result down to the nearest integer (4.5 rounds down to 4), and `%` is the modulus operator, which returns the remainder of dividing 9 by 2 (which is 1).

6.

```
num = 1

while num <= 10:
    print(num)
    num += 1
```

In this code, we first set the variable `num` to 1, since we want to print the first 10 natural numbers starting from 1. Then we use a while loop that will continue executing as long as `num` is less than or equal to 10.

Inside the loop, we simply print the current value of `num` using the `print()` function, and then we increment `num` by 1 using the `+=` operator. This ensures that the loop will eventually terminate after 10 iterations.

```
1
2
3
4
5
6
7
8
9
10
```

7.

```
num = int(input("Enter a number: "))

sum = 0

for i in range(1, num+1):
    sum += i

print("The sum of all numbers from 1 to", num, "is:",
```

In this code, we first use the `input()` function to accept a number from the user as input, and we store it in the variable `num`. Since the input is read as a string, we use the `int()` function to convert it to an integer.

Next, we initialize a variable `sum` to 0, which we will use to keep track of the running sum of numbers from 1 to `num`.

We then use a `for` loop to iterate from 1 to `num`, and for each iteration, we add the current value of `i` to the running sum `sum`.

Finally, we use the `print()` function to output the sum of all numbers from 1 to `num` to the console. For exemple

```
 The sum of all numbers from 1 to 10 is: 55
```

8.
```
for i in range(1, 51):
    if i % 3 == 0 and i % 5 == 0:
        print("FizzBuzz")
    elif i % 3 == 0:
        print("Fizz")
    elif i % 5 == 0:
        print("Buzz")
    else:
        print(i)
```

In this code, we use a `for` loop to iterate over the integers from 1 to 50, and for each integer, we use a series of `if` and `elif` statements to determine whether it is a multiple of 3, 5, or both.

If the integer is a multiple of both 3 and 5 (i.e., it is divisible by 15), we print "FizzBuzz" to the console using the `print()` function.

If the integer is only a multiple of 3, we print "Fizz" to the console. Similarly, if the integer is only a multiple of 5, we print "Buzz" to the console.

If the integer is not a multiple of 3 or 5, we simply print the integer itself.

```
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
16
17
Fizz
19
Buzz
Fizz
22
23
Fizz
```

```
Buzz
26
Fizz
28
29
FizzBuzz
31
32
Fizz
34
Buzz
Fizz
37
38
Fizz
Buzz
41
Fizz
43
44
FizzBuzz
46
47
Fizz
49
```

```
Buzz
```