Storage Variables

```
pragma solidity ^0.8.0;

contract Contract {
    bool public a = true;
    bool public b = false;
}
```

Unsigned Integers

```
pragma solidity ^0.8.0;

contract Contract {
    uint8 public a = 100;       // value between 0 and 255
    uint16 public b = 300;      // value at least 256
    uint256 public sum = a + b; // sum of a and b
}
```

Signed Integers

```
pragma solidity ^0.8.0;

contract Contract {
    int8 public a = 10;          // positive int8
    int8 public b = -15;         // negative int8
    int16 public difference = a - b; // 10 - (-15) = 25
}
```

String Literals

```
pragma solidity ^0.8.0;

contract Contract {
```

```
    bytes32 public msg1 = "Hello World";
    string public msg2 = "This message is longer than thirty-two bytes!";
}
```

Enum Type

```
pragma solidity ^0.8.0;

contract Contract {
    enum Foods { Pizza, Sushi, Burger, Tacos }

    Foods public food1 = Foods.Pizza;
    Foods public food2 = Foods.Sushi;
    Foods public food3 = Foods.Burger;
    Foods public food4 = Foods.Tacos;
}
```

Solidity Arguments

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract Contract {
    uint public x;

    constructor(uint _x) {
        x = _x;
    }
}
```

Contract Functions

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract Contract {
    uint public x;

    constructor(uint _x) {
        x = _x;
    }

    function increment() external {
        x += 1;
    }
}
```