**Random numbers - Block Timestamp**

**Prerequisites for picking a winner**

Going back to [lesson 1](https://updraft.cyfrin.io/courses/foundry/smart-contract-lottery/setup), we established that one of the Raffle contract goals is ...we should be able to automatically pick a winner out of the registered users.

What do we need to do that?

1. A random number
2. Use the random number to pick a winning player
3. Call pickWinner automatically

For now, let's focus on points 1 and 2. But before diving straight into the randomness let's think a bit about the Raffle design. We don't have any problem with anyone calling pickWinner. As long as someone wants to pay the gas associated with that they are more than welcome to do it. But we need to make sure that a decent amount of time passed since the start of the raffle. We don't want to host a 10-second raffle where two people get to register and then someone calls the pickWinner. In that sense, we need to define a new state variable called i\_interval which represents the duration of a raffle:

contract Raffle {

error Raffle\_\_NotEnoughEthSent();

uint256 private immutable i\_entranceFee;

// @dev Duration of the lottery in seconds

uint256 private immutable i\_interval;

address payable[] private s\_players;

event EnteredRaffle(address indexed player);

constructor(uint256 entranceFee, uint256 interval) {

i\_entranceFee = entranceFee;

i\_interval = interval;

}

}

Now that we have defined a raffle duration, we need to check it in pickWinner, but check it against what? We need to check it against the difference between the moment in time when the raffle started and the moment in time when the function pickWinner is called. But for that, we need to record the raffle starting time.

Perform the following update:

contract Raffle{

error Raffle\_\_NotEnoughEthSent();

uint256 private immutable i\_entranceFee;

// @dev Duration of the lottery in seconds

uint256 private immutable i\_interval;

address payable[] private s\_players;

uint256 private s\_lastTimeStamp;

event EnteredRaffle(address indexed player);

constructor(uint256 entranceFee, uint256 interval) {

i\_entranceFee = entranceFee;

i\_interval = interval;

s\_lastTimeStamp = block.timestamp;

}

}

And now we have all the prerequisites to perform the check:

// 1. Get a random number

// 2. Use the random number to pick a player

// 3. Automatically called

function pickWinner() external {

// check to see if enough time has passed

if (block.timestamp - s\_lastTimeStamp < interval) revert();

}

Don't worry! We will create a custom error for that in the next lesson. But before that let's talk randomness.