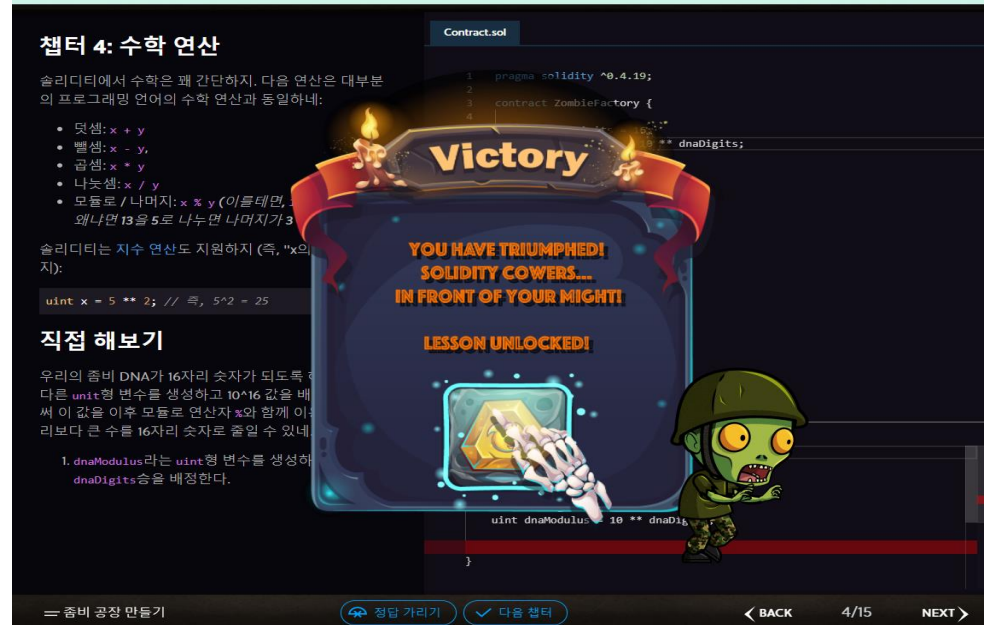
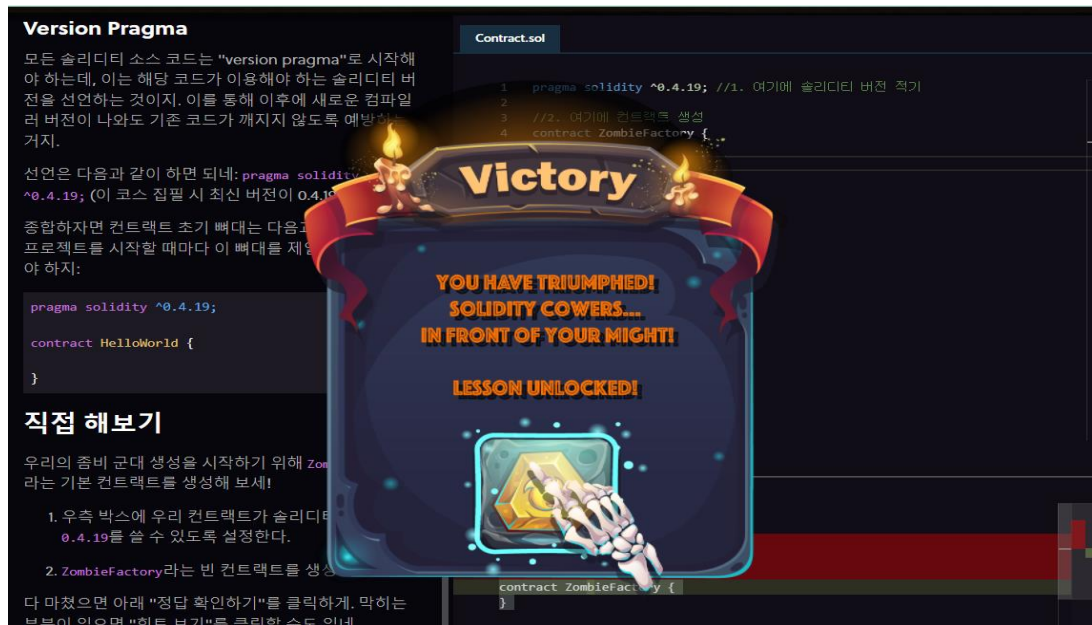
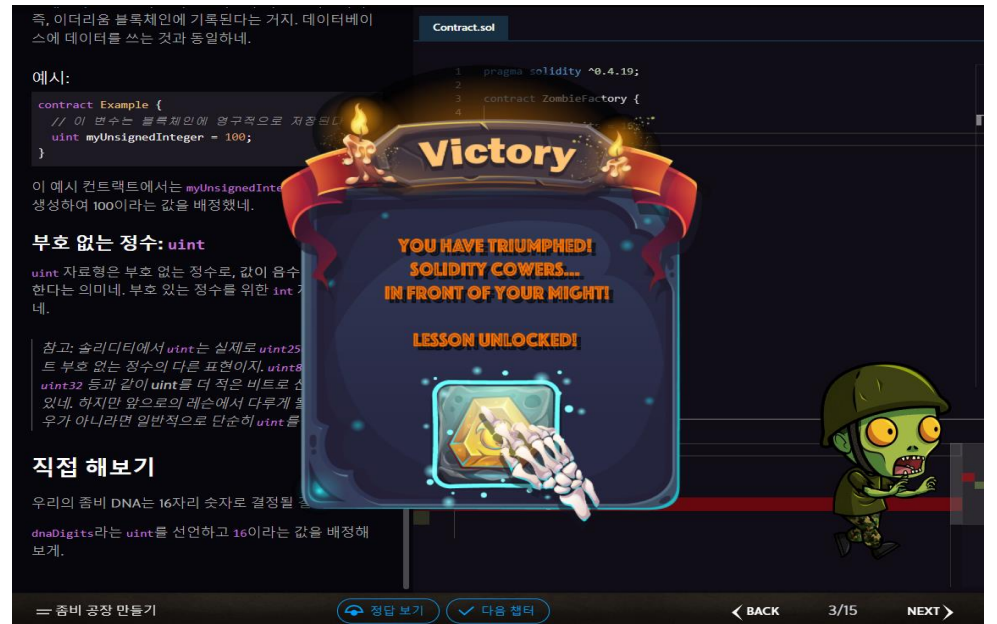


블록체인 과제 3

20155137

안원영



한빛SmartLEAD

CryptoZombies | Lesson 1 Cha

cryptozombies.io/lesson/1/chapter/5

YouTubeDateFormat class ~...antialcohol - 개요...Flutter 입문 - 인트...Cloud Firestore...Flutter Example ~...Remix - Ethereum L...

Your account has been successfully registered.

Contract.sol

```
1 pragma solidity ^0.4.19;
2
3 contract ZombieFactory {
4
5     uint public dnaDigits;
6
7     struct Person {
8         uint age;
9         string name;
10     }
11
12     Person[] public people;
13
14     function createZombie(string _name, uint _dna) public {
15         require(_dna < 2 ** dnaDigits);
16         people.push(Person(_name, _dna));
17     }
18 }
```

챕터 5: 구조체

자네가 좀 더 복잡한 자료형을 필요로 할 때가 가끔 있을 거네. 이를 위해 솔리디티는 구조체를 제공하지:

```
struct Person {
    uint age;
    string name;
}
```

구조체를 통해 여러 특성을 가진, 보다 복잡하게 생성할 수 있지.

참고로 *string*이라는 새로운 자료형을 받아들이는 임의의 길이를 가진 UTF-8 데이터 타입을 사용해. 이를테면 *string greeting = "Hello, world!"* 이렇게 말이야.

직접 해보기

우리 앱에서 좀비 몇 마리를 생성하기 위해 *Person* 구조체가 다양한 특성을 가질 것이니 구조체를 *Person*에 작성해보자.

- Zombie*라는 *struct*를 생성한다.
- 우리의 *Zombie* 구조체는 *name* (*string*형)과 *dna* (*uint*형)이라는 2가지 특성을 가진다.

== 좀비 공장 만들기

정답 보기

다음 챕터

BACK5/15NEXT

한빛SmartLEAD

CryptoZombies | Lesson 1 Cha

cryptozombies.io/lesson/1/chapter/7

YouTubeDateFormat class ~...antialcohol - 개요...Flutter 입문 - 인트...Cloud Firestore...Flutter Example ~...Remix - Ethereum L...

Your account has been successfully registered.

Contract.sol

```
1 pragma solidity ^0.4.19;
2
3 contract ZombieFactory {
4
5     uint public dnaDigits;
6
7     struct Person {
8         uint age;
9         string name;
10     }
11
12     Person[] public people;
13
14     function createZombie(string _name, uint _dna) public {
15         require(_dna < 2 ** dnaDigits);
16         people.push(Person(_name, _dna));
17     }
18 }
```

챕터 7: 함수 선언

솔리디티에서 함수 선언은 다음과 같이 해:

```
function eatHamburgers(string _name, uint _amount) public {
    // ...
}
```

이 함수는 *eatHamburgers*라는 함수로, *string*과 *uint*을 인자로 전달받고 있고, 현재로서는 함수의 내용이 비어있어.

참고: 함수 인자명을 언더스코어(`_`)로 시작해서 구별하는 것이 관례야 (의무는 아님). 본 튜토리얼에서 이 관례를 따를 것이네.

이 함수를 다음과 같이 호출할 수 있지:

```
eatHamburgers("vitalik", 100);
```

직접 해보기

우리 앱에서 좀비들을 생성할 수 있을 필요가 있을 것 같아. *Person* 구조체를 생성해 보자.

- createZombie*라는 함수를 생성한다. 이 함수는 *string*과 *uint*의 인자를 전달받아야 한다: *_name* (*string*형)과 *_dna* (*uint*형).

함수의 내용은 지금으로선 비어 두면 되네. 나중에 채울 것이니까.

== 좀비 공장 만들기

정답 가리기

다음 챕터

BACK7/15NEXT

한빛SmartLEAD

CryptoZombies | Lesson 1 Cha

cryptozombies.io/lesson/1/chapter/6

YouTubeDateFormat class ~...antialcohol - 개요...Flutter 입문 - 인트...Cloud Firestore...Flutter Example ~...Remix - Ethereum L...

Your account has been successfully registered.

Contract.sol

```
1 pragma solidity ^0.4.19;
2
3 contract ZombieFactory {
4
5     uint public dnaDigits;
6
7     struct Person {
8         uint age;
9         string name;
10     }
11
12     Person[] public people;
13
14     function createZombie(string _name, uint _dna) public {
15         require(_dna < 2 ** dnaDigits);
16         people.push(Person(_name, _dna));
17     }
18 }
```

Public 배열

*public*으로 배열을 선언할 수 있지. 솔리디티에서 배열을 위해 *getter* 메소드를 자동으로 생성해 주는데, 다음과 같네:

```
Person[] public people;
```

그러면 다른 컨트랙트들이 이 배열을 읽을 수 있게 되는데 (읽 수는 없네). 이는 컨트랙트에 공개 데이터가 필요할 때 유용한 패턴이지.

직접 해보기

우리 앱에 좀비 군대를 저장하고 싶네. 그러려면 *Person* 구조체를 *public*으로 선언하고 싶네. 그러려면 *Person* 구조체를 *public*으로 선언하고 싶네. 그러려면 *Person* 구조체를 *public*으로 선언하고 싶네.

- Zombie* 구조체의 *public* 배열을 생성하고 이름을 *zombies*로 한다.

== 좀비 공장 만들기

정답 보기

다음 챕터

BACK6/15NEXT

한빛SmartLEAD

CryptoZombies | Lesson 1 Cha

cryptozombies.io/lesson/1/chapter/8

YouTubeDateFormat class ~...antialcohol - 개요...Flutter 입문 - 인트...Cloud Firestore...Flutter Example ~...Remix - Ethereum L...

Your account has been successfully registered.

Contract.sol

```
1 pragma solidity ^0.4.19;
2
3 contract ZombieFactory {
4
5     uint public dnaDigits;
6
7     struct Person {
8         uint age;
9         string name;
10     }
11
12     Person[] public people;
13
14     function createZombie(string _name, uint _dna) public {
15         require(_dna < 2 ** dnaDigits);
16         people.push(Person(_name, _dna));
17     }
18 }
```

이제 새로운 Person을 생성하고 people 배열에 추가하는 방법을 살펴보자.

// 새로운 사람을 생성한다:
Person satooshi = Person(172, "Satoshi");

// 이 사람을 배열에 추가한다:
people.push(satooshi);

이 두 코드를 조합하여 깔끔하게 한 줄로 표현할 수 있지:

```
people.push(Person(16, "Vitalik"));
```

참고로 *array.push()*는 무언가를 배열의 끝에 추가하는 메소드이지. 원소를 추가하도록 하네. 다음 예시를 살펴보자:

```
uint[] numbers;
numbers.push(5);
numbers.push(10);
numbers.push(15);
// numbers 배열은 [5, 10, 15]과 같다.
```

직접 해보기

createZombie 함수가 무언가를 할 수 있도록 만들어 보자.

- 함수에 코드를 넣어 새로운 *Zombie*를 생성하고 *people* 배열에 추가하도록 한다. 새로운 좀비를 위한 *name*과 *dna*는 *createZombie*함수의 인자값이어야 한다.
- 코드를 한 줄로 간결하게 작성해 보자.

== 좀비 공장 만들기

정답 가리기

다음 챕터

BACK8/15NEXT

한빛SmartLEAD

CryptoZombies | Lesson 1 Chap

cryptozombies.io/lesson/1/chapter/10

YouTubeDateFormat class ~...antialcohol ~ 개요...Flutter 입문 - 안드...Cloud Firestore로...Flutter Example ~...Remix - Ethereum L...읽기 목록

Your account has been successfully registered.

Contract.sol

```
function _multiply(uint a, uint b) private pure returns (uint) {
    return a * b;
}

uint dnaModulus = 10 ** dnaDigits;
struct Zombie {
    string name;
    uint dna;
}

(string _name, uint _dna) private {
    _dna);
}

(string _str) private view returns (uint) {
    256(_str));
};
}

== 준비 공장 만들기 ==
```

Victory

YOU HAVE TRIUMPHED!
SOLIDITY COWERS...
IN FRONT OF YOUR MIGHT!

LESSON UNLOCKED!

== 준비 공장 만들기 ==

항답 보기

다음 섹터

BACK

10/15

NEXT

솔리디티는 **pure** 함수도 가지고 있는데, 이는 함수가 랩에서 어떤 데이터도 접근하지 않는 것을 의미하지. 다음을 살펴보기:

```
function _multiply(uint a, uint b) private pure returns (uint) {
    return a * b;
}

이 함수는 랩에서 읽는 것도 하지 않고, 다만 반환된 전달된 인자값에 따라서 달라지지만, 그러니 이 경우 pure로 선언하지.

참고: 함수를 pure나 view로 언제 표시할지 기억할 수 있지. 온 종게도 솔리디티 컴파일러는 어떤 제어를 하지 않고 메시지를 통해 할 알려주네.



### 직접 해보기



스트링으로부터 랜덤 DNA 숫자를 생성하는 도우미 요할 결제.



1. _generateRandomDna라는 private 함수를 만드는 _str(string형)을 인자로 전달받고, uint야 하네.
2. 이 함수는 컨트랙트 변수를 보지만 변경하지 않으므로 view로 선언하게.
3. 이 함수의 내용은 현재로서는 비어 있는 상태로 넘두시게. 나중에 채울 것이네.

```

한빛SmartLEAD

CryptoZombies | Lesson 1 Chap

cryptozombies.io/lesson/1/chapter/11

YouTubeDateFormat class ~...antialcohol ~ 개요...Flutter 입문 - 안드...Cloud Firestore로...Flutter Example ~...Remix - Ethereum L...읽기 목록

Your account has been successfully registered.

Contract.sol

```
pragma solidity ^0.4.10;
contract ZombieFactory {
    uint8 a = 5;
    uint8 b = 6;
    // a * b? uint8이 아닌 uint를 반환하기 때문에
    uint8 c = a * b;
    // b를 uint8으로 형 변환해서 코드가 제대로
    uint8 c = a * uint8(b);
}

위의 예시에서 a * b는 uint를 반환하지. 하지만 우린 값을 uint8에 저장하려고 하니 잠재적으로 문제가 있네. 반환값을 uint8으로 형 변환하면 코드가 제대로 컴파일러도 에러 메시지를 주지 않을 걸세.

### 직접 해보기



_generateRandomDna 함수의 내용을 채워 보세! 여기 무엇을 해야 하는지 나와 있네:



1. 코드 첫 줄에서는 _str을 이용한 keccak256 해시에서 의사 난수 16진수를 생성하고 이를 uint8다음, rand라는 uint에 결과값을 저장해야 한다.
2. 우리는 종비의 DNA가 16자리 숫자이기만을 (dnaModulus를 기억하거나) 코드의 두번째 줄에서는 위의 결과 값을 모듈로(%) dnaModulus로 연산한 값을 반환해야 한다. (형식: return 위의 결과 값 % dnaModulus).

```

Victory

YOU HAVE TRIUMPHED!
SOLIDITY COWERS...
IN FRONT OF YOUR MIGHT!

LESSON UNLOCKED!

== 준비 공장 만들기 ==

항답 보기

다음 섹터

BACK

11/15

NEXT

한빛SmartLEAD

CryptoZombies | Lesson 1 Chap

cryptozombies.io/lesson/1/chapter/9

YouTubeDateFormat class ~...antialcohol ~ 개요...Flutter 입문 - 안드...Cloud Firestore로...Flutter Example ~...Remix - Ethereum L...읽기 목록

Your account has been successfully registered.

Contract.sol

```
pragma solidity ^0.4.10;
contract ZombieFactory {
    uint8 a = 5;
    uint8 b = 6;
    // a * b? uint8이 아닌 uint를 반환하기 때문에
    uint8 c = a * b;
    // b를 uint8으로 형 변환해서 코드가 제대로
    uint8 c = a * uint8(b);
}

uint[] numbers;

function _addToArray(uint _number) private {
    numbers.push(_number);
}

private는 컨트랙트 내의 다른 함수들만이 이 함수를 numbers 배열로 무언가를 추가할 수 있다는 것을 의미. 위의 예시에서 볼 수 있듯이 private 키워드는 함수적네. 함수 인자명과 마찬가지로 private 함수명도 시작하는 것이 관례야네.

### 직접 해보기



우리 컨트랙트의 createZombie 함수는 현재 기본적으로 선언되어 있네. 즉, 누구나 이 함수를 호출해서 비를 컨트랙트에서 만들 수 있다는 뜻이지! 이 함수로 선언해 보세.



1. createZombie 함수를 변경하여 private 함수로 만든 수명에 대한 관례를 잊지 말 것!

```

Victory

YOU HAVE TRIUMPHED!
SOLIDITY COWERS...
IN FRONT OF YOUR MIGHT!

LESSON UNLOCKED!

== 준비 공장 만들기 ==

항답 보기

다음 섹터

BACK

9/15

NEXT

한빛SmartLEAD

CryptoZombies | Lesson 1 Chap

cryptozombies.io/lesson/1/chapter/12

YouTubeDateFormat class ~...antialcohol ~ 개요...Flutter 입문 - 안드...Cloud Firestore로...Flutter Example ~...Remix - Ethereum L...읽기 목록

Your account has been successfully registered.

Contract.sol

```
uint dnaModulus = 10 ** dnaDigits;
struct Zombie {
    string name;
    uint dna;
}

(string _name, uint _dna) private {
    _dna);
}

(string _str) private view returns (uint) {
    256(_str));
};
}

== 준비 공장 만들기 ==
```

Victory

YOU HAVE TRIUMPHED!
SOLIDITY COWERS...
IN FRONT OF YOUR MIGHT!

LESSON UNLOCKED!

== 준비 공장 만들기 ==

항답 보기

다음 섹터

BACK

12/15

NEXT

챕터 12: 종합하기

랜덤 종비 생성기를 거의 다 완성해 가는군! 이제 모든 내용을 종합하는 `public` 함수를 생성해 보세.

종비의 이름을 입력값으로 받아 랜덤 DNA를 가진 종비를 `public` 함수를 생성할 결제.

직접 해보기

1. `createRandomZombie`라는 `public` 함수를 생성하는 `_name`이라는 `string`형 인자를 하나 전달한 함수를 `private`로 선언한 것과 마찬가지로 `uint`로 생성할 것
2. 이 함수의 첫 줄에서는 `_name`을 전달받은 `_generateRandomDna` 함수를 호출하고, 이 함수를 `randDna`라는 `uint`형 변수에 저장해야 한다.
3. 두번째 줄에서는 `_createZombie` 함수를 호출하여 `_name`과 `randDna`를 전달해야 한다.
4. 함수의 내용을 닫는 `}`를 포함해서 코드가 4줄이다.

