

YAML

<https://youtu.be/4NFV4vE6JC8>

YAML

- Yet Another Markup Language → YAML Ain't Markup Language
- XML이나 JSON처럼 데이터를 나타내는데 사용
- 사람이 읽기에 XML이나 JSON보다 더 적합
- .yaml, .yml
- 한국에서는 주로 '야믈'로 발음

YAML

- 파이썬과 같이 들여쓰기를 통해 값 구분

XML	JSON	YAML
<pre><Servers> <Server> <name>Server1</name> <owner>John</owner> <created>123456</created> <status>active</status> </Server> </Servers></pre>	<pre>{ Servers: [{ name: Server1, owner: John, created: 123456, status: active }] }</pre>	<pre>Servers: - name: Server1 owner: John created: 123456 status: active</pre>

YAML

- 모든 값은 key: value 형태로 표현
 - : 이후에는 한 칸 공백
 - ex) year: 2021
- 스칼라 (scalar)
 - 문자열, 숫자, boolean
- 시퀀스 (sequence)
 - 배열
- 맵핑 (mapping)
 - 해시, 딕셔너리, 키-값

YAML

- 파이썬과 같이 들여쓰기 글자수가 일정해야 함

```
key:
  key1:
    key2:
      key3:
        ...
```

- JSON과 달리 YAML은 주석이 있음
 - 주석은 #으로 표현
 - # 이것은 주석입니다.

YAML - 자료형

- number_type: 10
- string_type: hello
 - 강제로 문자열로 입력받기 위해서는 작은따옴표(')나 큰따옴표("") 사용
 - id: "20213201"
 - 특수문자 및 escape code의 처리를 위해서는 큰따옴표, 원치 않으면 작은 따옴표 사용
 - control: "\b1998\t1999\t2000\n"
- boolean_type: true

YAML - object

- key:
 key1: value
 key2: value
- `또는`
- key: {
 key1: value
 key2: value
}

YAML - 배열

- key:
 - item
 - item
- 또는
- key: [
 - item, item]

YAML - 해시

name: John Smith

age: 33

또는

{name: John Smith, age: 33}

YAML - 해시와 리스트

해시의 리스트

- [name: John Smith, age: 33]

또는

- name: John Smith
age: 2

YAML - 해시와 리스트

리스트의 해시

```
names: [John, Bill]
```

또는

```
names:
```

- John
- Bill

YAML

--- : 문서의 시작 (생략 가능)

... : 문서의 끝 (생략 가능)

ex)

name: John

job: developer

...

YAML

문자열 표현에 있어서, |는 줄바꿈을 포함하고 >는 줄바꿈 무시

```
comment1: |  
  Lorem ipsum  
  dolor sit amet
```

```
comment2: >  
  Lorem ipsum  
  dolor sit amet
```

```
# comment1: Lorem ipsum\n  dolor sit amet\n# comment2: Lorem ipsum dolor sit amet
```

YAML

newlines preserved

--- |

There was a young fellow of Warwick
Who had reason for feeling euphoric
For he could, by election
Have triune erection
Ionic, Corinthian, and Doric 11

newlines folded

--- >

Wrapped text
will be folded
into a single
paragraph

Blank lines denote
paragraph breaks

YAML - &, *, <<

- & : 앵커 (anchor)
- * : 에일리어스 (alias)
- << : 상속

YAML - &, *, <<

variables pre-define

colors:

- color1: &color1 yellow
- color2: &color2 green

frame:

title: *color1
logo: *color2

article:

title: *color2
content: *color1

YAML - &, *, <<

```
person: &default  
  name: John Doe  
  color: black
```

==

```
person: &default  
  name: John Doe  
  color: black
```

```
employee: &employee  
  <<: *default  
  salary: 500
```

```
employee: &employee  
  name: John Doe  
  color: black  
  salary: 500
```

YAML - &, *, <<

```
person: &person  
  name: John Doe  
  color: black
```

```
employee: &employee
```

```
  <<: *person
```

```
  name: Bill
```

← 'name' 필드를 오버라이딩 하여 'Bill'로 만듦

```
  salary: 500
```

YAML - 구현

- 자바스크립트
- Object-C
- 펄
- PHP
- 파이썬
- 루비
- 자바
- 하스켈
- XML
- 러스트

YAML with Docker

- 도커에서는 설정 파일을 YAML로 관리
- yaml 파일을 통해 컨테이너 생성 및 서비스 설정이 가능
- <https://bit.ly/2Y7Az3V>

YAML with Docker - 버전

- 파일 맨 윗부분에 버전 명시
- ex) version: “3”

YAML with Docker - 서비스

- 서비스를 통해 도커 컴포즈로 생성할 컨테이너의 옵션을 정의
- 각 서비스는 컨테이너로 구현됨
- 서비스 컨테이너를 생성할 때 사용할 이미지를 설정할 수 있음

services:

my_container_1:

image: ubuntu18.04

my_container_2:

image: nginx

YAML with Docker - 서비스

- 서비스는 image 이외에도 다양한 옵션들을 포함
 - links: 문자열 배열, 다른 서비스에 서비스명을 통해 접근 설정
 - environment: 문자열 배열, 환경변수를 지정하여 딕셔너리나 배열 형태로 사용
 - command: 문자열 배열 or 문자열, 컨테이너가 실행될 때 수행할 명령어
 - ports: 문자열 배열, 서비스 컨테이너에 개방할 포트 설정
 - volume: 문자열 배열, 컨테이너의 데이터를 저장하기 위해 사용

YAML with Docker

- fabric-samples/test-network/docker/docker-compose-ca.yaml

```
version: '2'

networks:
  test:

services:
  ca_org1:
    image: hyperledger/fabric-ca:$IMAGE_TAG
    environment:
      - FABRIC_CA_HOME=/etc/hyperledger/fabric-ca-server
      - FABRIC_CA_SERVER_CA_NAME=ca-org1
      - FABRIC_CA_SERVER_TLS_ENABLED=true
      - FABRIC_CA_SERVER_PORT=7054
    ports:
      - "7054:7054"
    command: sh -c 'fabric-ca-server start -b admin:adminpw -d'
    volumes:
      - ../organizations/fabric-ca/org1:/etc/hyperledger/fabric-ca-server
    container_name: ca_org1
    networks:
      - test
```

CA 조직 'ca_org1'의 생성을 위한 설정

YAML with Hyperledger Fabric

```
Organizations:
- &OrdererOrg
  Name: OrdererOrg
  ID: OrdererMSP
  MSPDir: ../organizations/ordererOrganizations/hn.com/msp
  Policies:
    Readers:
      Type: Signature
      Rule: "OR('OrdererMSP.member')"
    Writers:
      Type: Signature
      Rule: "OR('OrdererMSP.member')"
    Admins:
      Type: Signature
      Rule: "OR('OrdererMSP.admin')"
  OrdererEndpoints:
    - orderer.hn.com:7050

- &Org1
  Name: Org1MSP
  ID: Org1MSP
  MSPDir: ../organizations/peerOrganizations/org1.hn.com/msp
  Policies:
    Readers:
      Type: Signature
      Rule: "OR('Org1MSP.admin', 'Org1MSP.peer', 'Org1MSP.client')"
    Writers:
      Type: Signature
      Rule: "OR('Org1MSP.admin', 'Org1MSP.client')"
    Admins:
      Type: Signature
      Rule: "OR('Org1MSP.admin')"
    Endorsement:
      Type: Signature
      Rule: "OR('Org1MSP.peer')"
  AnchorPeers:
    - Host: peer0.org1.hn.com
      Port: 7051
```

오더러와 'Org1' 조직의 구성에 대한 설정 파일
(configtx.yaml)

Q & A

