

# 기본 틀

```
rule myrule:  
    input: "path/to/{sample}.txt"  
    output: "path/to/{sample}.column1.txt"  
    shell: "cut -f1 < {input} > {output}"
```

# input, output에 변수 설정

```
rule myrule:  
    input: a="path/to/{sample}.txt"  
    output: b="path/to/{sample}.column1.txt"  
    shell: "cut -f1 < {input.a} > {output.b}"
```

shell : 웰 스크립트

run : Python

```
rule myrule:  
    input: a="path/to/{sample}.txt"  
    output: b="path/to/{sample}.column1.txt"  
    run:  
        with open(output.b, "w") as out:  
            for l in csv.reader(open(input.a)):  
                print(l[0], file=out)
```

# Thread

```
rule myrule:  
    input: "path/{sample}.txt"  
    output: "path/{sample}.sorted.txt"  
    threads: 8  
    shell: "sort --parallel={threads} {input} > {output}"
```

Default rule

or

최종 rule

# Dependencies 자동 계산

“File-driven  
programming”

```
# require a bam for each sample
rule all:
    input: "500.bam", "501.bam", "502.bam", "503.bam"

# map reads
rule map:
    input:
        ref="reference.fasta", index="reference.bwt",
        reads="{sample}.fastq"
    output: "{sample}.bam"
    threads: 8
    shell:
        "bwa mem -t {threads} {input.ref} {input.reads} | "
        "samtools view -Sbh - > {output}"

# create an index
rule index:
    input: "reference.fasta"
    output: "reference.bwt"
    shell: "bwa index {input}"
```

```
# require a bam for each sample
rule all:
    input: [str(no) + ".bam" for no in range(500, 504)]  
  
# map reads
rule map:
    input:
        ref="reference.fasta", index="reference.bwt",
        reads="{sample}.fastq"
    output: "{sample}.bam"
    threads: 8
    shell:
        "bwa mem -t {threads} {input.ref} {input.reads} | "
        "samtools view -Sbh - > {output}"  
  
# create an index
rule index:
    input: "reference.fasta"
    output: "reference.bwt"
    shell: "bwa index {input}"
```

Shell 부분을 제외  
하면 python 코드  
가능 영역

```
# require a bam for each sample
SAMPLE_OUTPUTS = [str(no) + ".bam" for no in range(500, 504)]
rule all:
    input: SAMPLE_OUTPUTS
```

```
# map reads
rule map:
    input:
        ref="reference.fasta", index="reference.bwt",
        reads="{sample}.fastq"
    output: "{sample}.bam"
    threads: 8
    shell:
        "bwa mem -t {threads} {input.ref} {input.reads} | "
        "samtools view -Sbh - > {output}"
```

```
# create an index
rule index:
    input: "reference.fasta"
    output: "reference.bwt"
    shell: "bwa index {input}"
```

Shell 부분을 제외  
하면 python 코드  
가능 영역

# 내장 함수 지원

## expand()

## shell()

## 등

```
# require a bam for each sample
SAMPLES = "500 501 502 503".split()
rule all:
    input: expand("{sample}.bam", sample=SAMPLES)

# map reads
rule map:
    input:
        ref="reference.fasta", index="reference.bwt",
        reads="{sample}.fastq"
    output: "{sample}.bam"
    threads: 8
    shell:
        "bwa mem -t {threads} {input.ref} {input.reads} | "
        "samtools view -Sbh - > {output}"

# create an index
rule index:
    input: "reference.fasta"
    output: "reference.bwt"
    shell: "bwa index {input}"
```

# 실행시 Python 코드로 번역후 실행

```
# require a bam for each sample
SAMPLES = "500 501 502 503".split()
@workflow.rule(name='all', lineno=3, snakefile='/atp/hyeshik/Snakefile')
@workflow.input(expand("{sample}.bam", sample=SAMPLES))

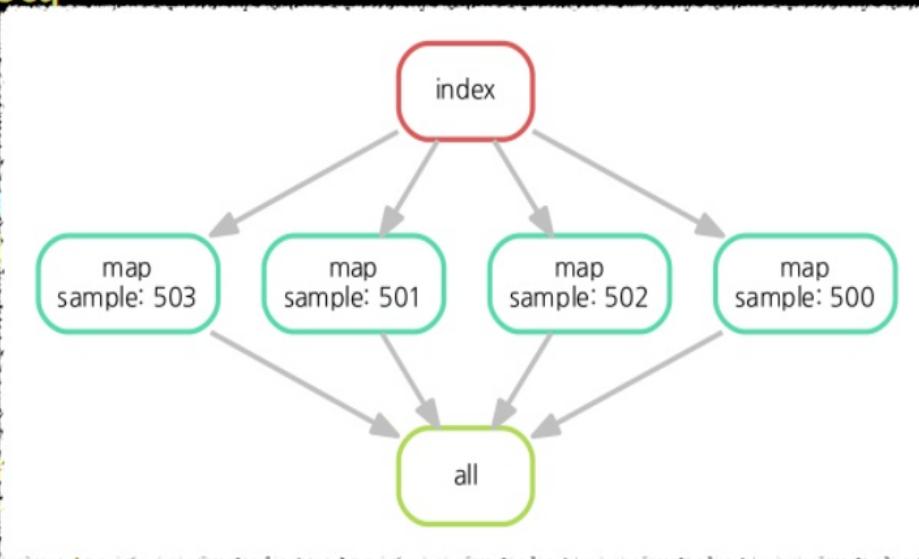
# map reads
@workflow.norun()
@workflow.run
def __all(input, output, params, wildcards, threads, resources, log, version):
    pass
@workflow.rule(name='map', lineno=13, snakefile='/atp/hyeshik/Snakefile')
@workflow.input(ref="reference.fasta", index="reference.bwt",
                reads="{sample}.fastq")
@workflow.output("{sample}.bam")
@workflow.threads(8)
@workflow.shellcmd("bwa mem -t {threads} {input.ref} {input.reads} | "
                  "samtools view -Sbh - > {output}")
@workflow.run
def __map(input, output, params, wildcards, threads, resources, log, version):
    shell("bwa mem -t {threads} {input.ref} {input.reads} | "
          "samtools view -Sbh - > {output}")

@workflow.rule(name='index', lineno=37, snakefile='/atp/hyeshik/Snakefile')
@workflow.input("reference.fasta")
@workflow.output("reference.bwt")
@workflow.shellcmd("bwa index {input}")
```

```
# require a bam for each sample
SAMPLES = "500 501 502 503".split()
rule all:
    input: expand("{sample}.bam", sample=SAMPLES)

# map reads
rule map:
    input:
        ref="reference.fasta", index="reference.bwt",
        reads="{sample}.fastq"
    output: "{sample}.bam"
    threads: 8
    shell:
        "bwa mem -t {threads} {index} {reads} | samtools view -Sb - > {output}"
        "samtools sort -m 2G {output} -o {output}.tmp"
        "mv {output}.tmp {output}"

# create an index
rule index:
    input: "reference.fasta"
    output: "reference.bwt"
    shell: "bwa index {input}"
```



## Directed Acycle Graph (DAG)

작업흐름도 생성