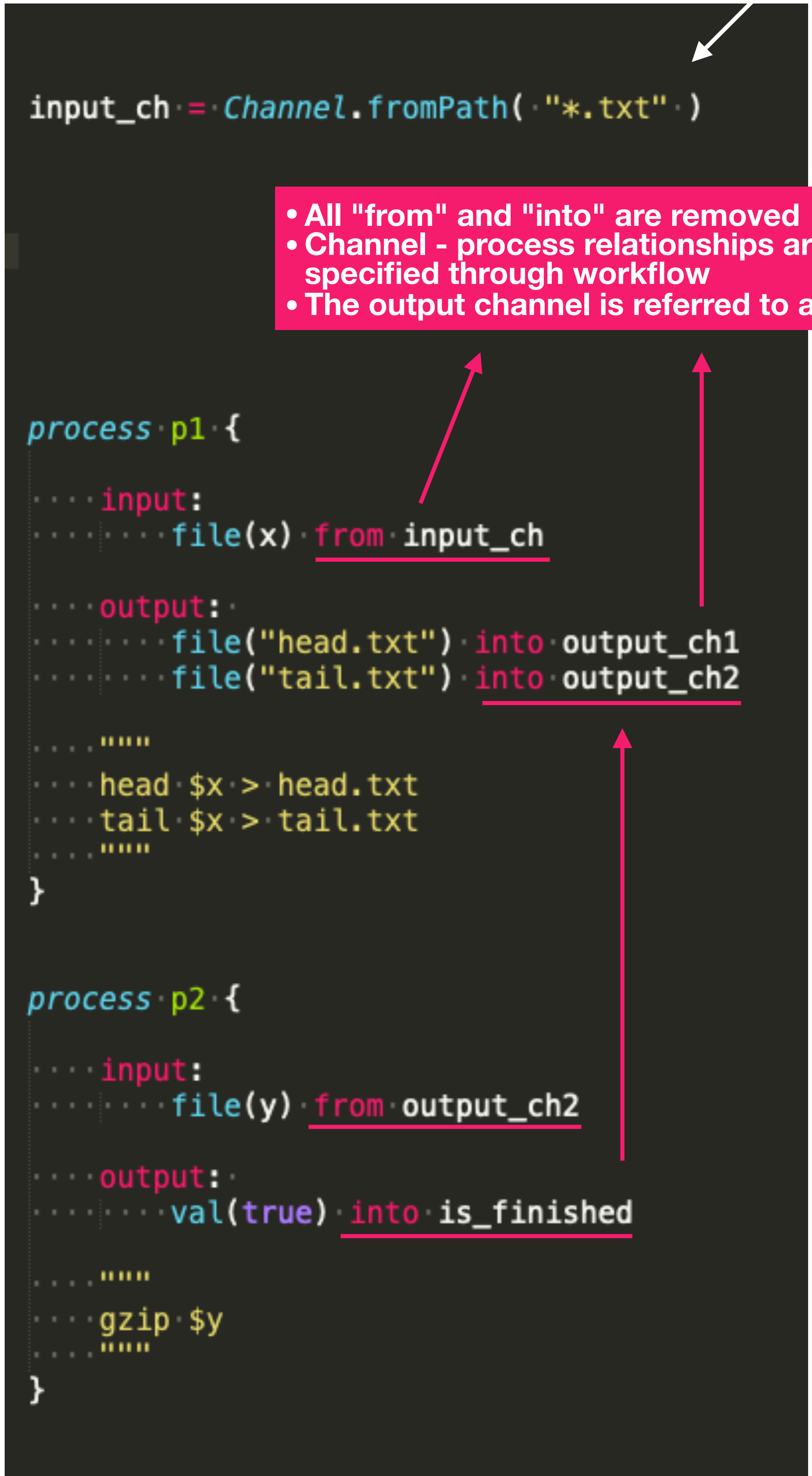
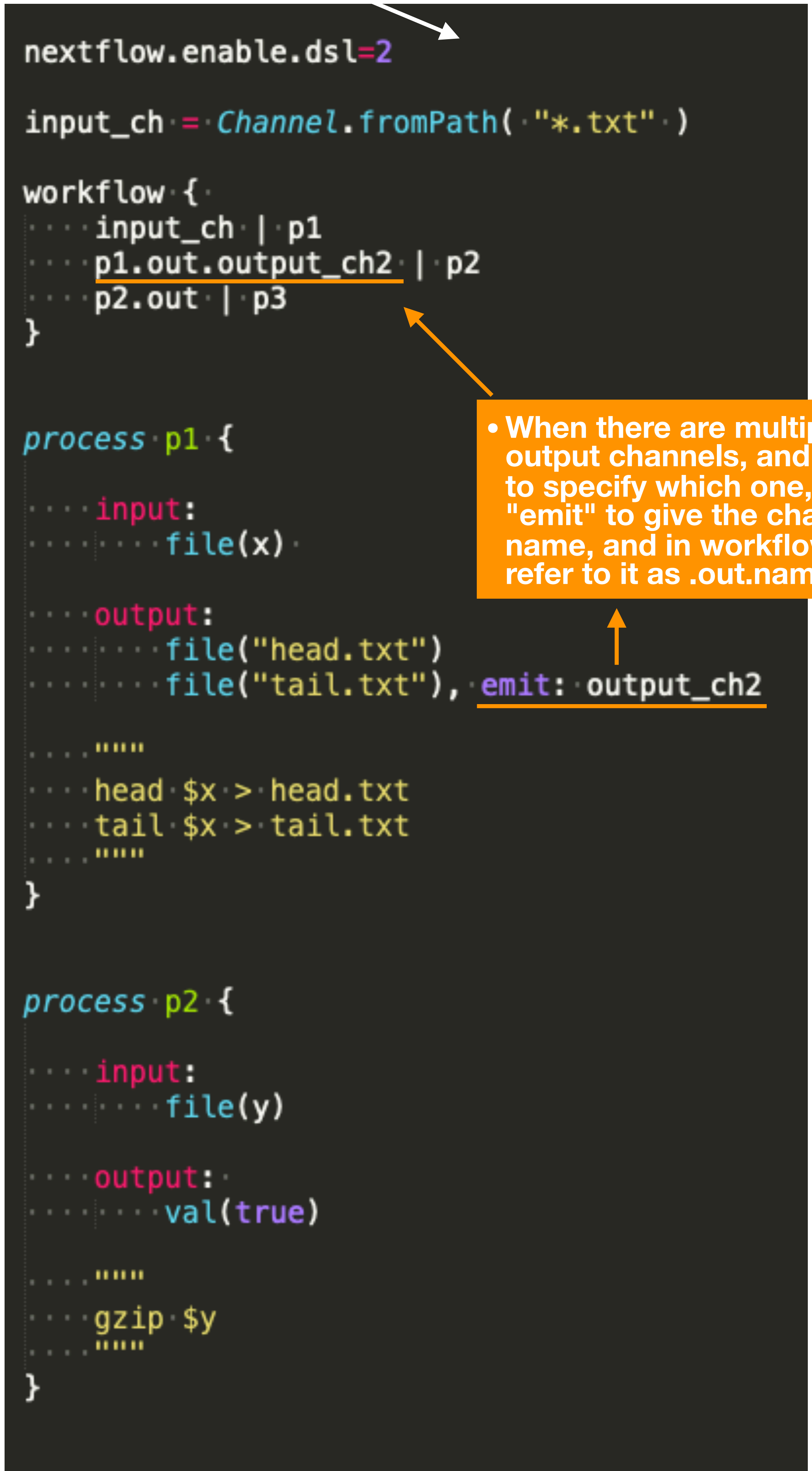
Nextflow: difference between the original syntax and DSL2



```
input_ch = Channel.fromPath( "*.txt" )


process p1 {

    input:
        file(x) from input_ch

    output:
        file("head.txt") into output_ch1
        file("tail.txt") into output_ch2

    """
    head $x > head.txt
    tail $x > tail.txt
    """
}


process p2 {

    input:
        file(y) from output_ch2

    output:
        val(true) into is_finished

    """
    gzip $y
    """
}
```

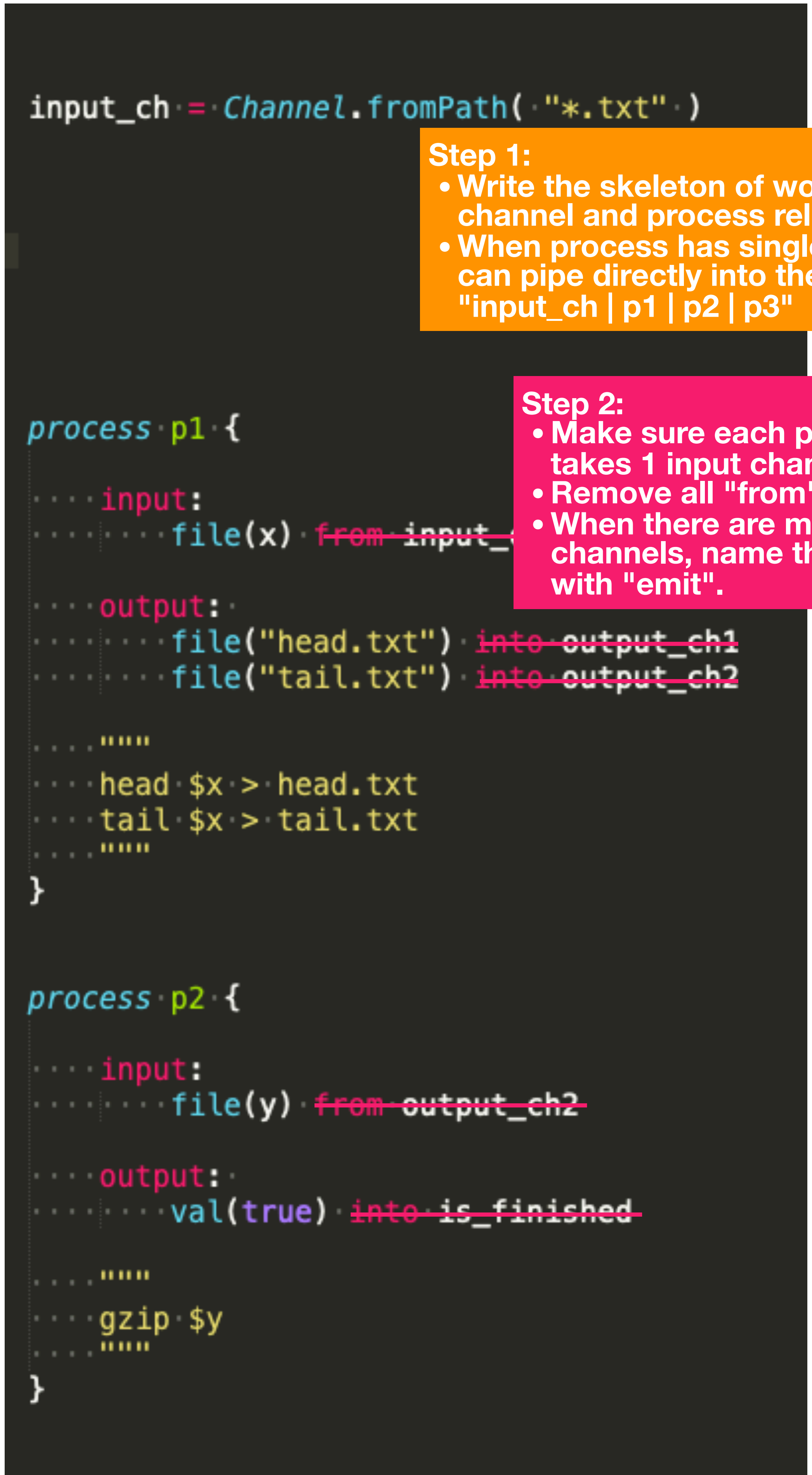
- All "from" and "into" are removed
- Channel - process relationships are now specified through workflow
- The output channel is referred to as p.out



```
nextflow.enable.dsl=2

input_ch = Channel.fromPath( "*.txt" )

workflow {
    input_ch | p1
    p1.out.output_ch2 | p2
    p2.out | p3
}


process p1 {

    input:
        file(x)

    output:
        file("head.txt")
        file("tail.txt"), emit: output_ch2

    """
    head $x > head.txt
    tail $x > tail.txt
    """
}


process p2 {

    input:
        file(y)

    output:
        val(true)

    """
    gzip $y
    """
}
```

- When there are multiple output channels, and need to specify which one, use "emit" to give the channel a name, and in workflow refer to it as .out.name

Practical steps:

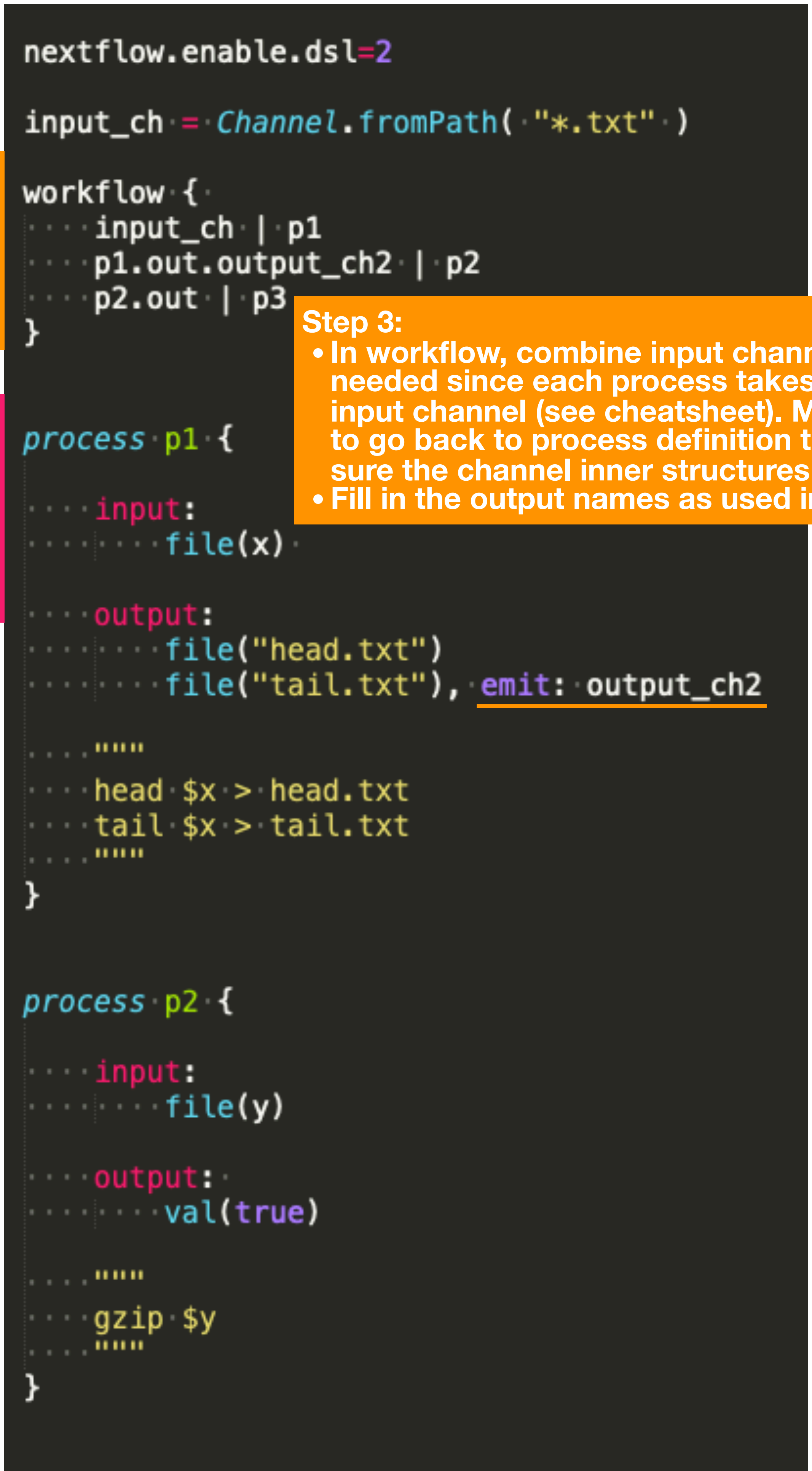


```
input_ch = Channel.fromPath( "*.txt" )



process p1 {

    input:
        file(x) ~~from input_~~

    output:
        file("head.txt") ~~into output_ch1~~
        file("tail.txt") ~~into output_ch2~~

    """
    head $x > head.txt
    tail $x > tail.txt
    """
}


process p2 {

    input:
        file(y) ~~from output_ch2~~

    output:
        val(true) ~~into is_finished~~

    """
    gzip $y
    """
}
```

Step 1:
- Write the skeleton of workflow based on channel and process relationships.
- When process has single output channel, can pipe directly into the next process: "input_ch | p1 | p2 | p3"

Step 2:
- Make sure each process only takes 1 input channel.
- Remove all "from" and "into".
- When there are multiple output channels, name them as needed with "emit".

DSL2



```
nextflow.enable.dsl=2

input_ch = Channel.fromPath( "*.txt" )

workflow {
    input_ch | p1
    p1.out.output_ch2 | p2
    p2.out | p3
}

process p1 {

    input:
        file(x)

    output:
        file("head.txt")
        file("tail.txt"), emit: output_ch2

    """
    head $x > head.txt
    tail $x > tail.txt
    """
}


process p2 {

    input:
        file(y)

    output:
        val(true)

    """
    gzip $y
    """
}
```

Step 3:
- In workflow, combine input channels as needed since each process takes only 1 input channel (see cheatsheet). May need to go back to process definition to make sure the channel inner structures agree.
- Fill in the output names as used in "emit".