

23강_node 모듈

- 23-1 .js파일 생성 및 실행
- 23-2 모듈이란?
- 23-3 직접만든 모듈
- 23-4 내부모듈
- 23-5 외부모듈

23-1 .js파일 생성 및 실행

The image illustrates the steps to create and run a JavaScript file in Eclipse IDE:

- File Creation:** A 'New JavaScript File' dialog box is shown. The 'Container' is set to '/pjt_HelloWorld' and the 'File name' is 'jsFile01.js'. The 'Finish' button is highlighted.
- Code Editing:** The main IDE window shows the 'jsFile01.js' file with the following code:


```

1  /**
2   * http://usejsdoc.org/
3   */
4
5  var temVar = 2;
6
7  for(var i = 1; i < 10; i++) {
8      console.log(temVar + " * " + i + " = " + (temVar*i));
9  }
10

```
- Execution:** The console window on the right shows the output of the code:

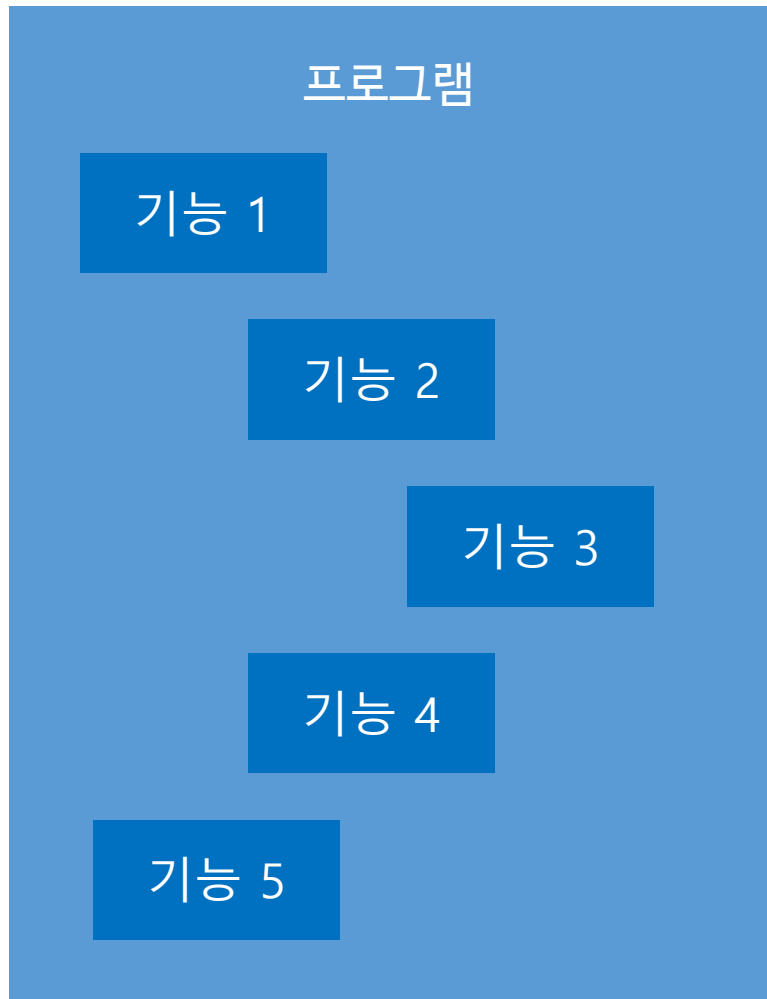

```

<terminated> pjt_
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18

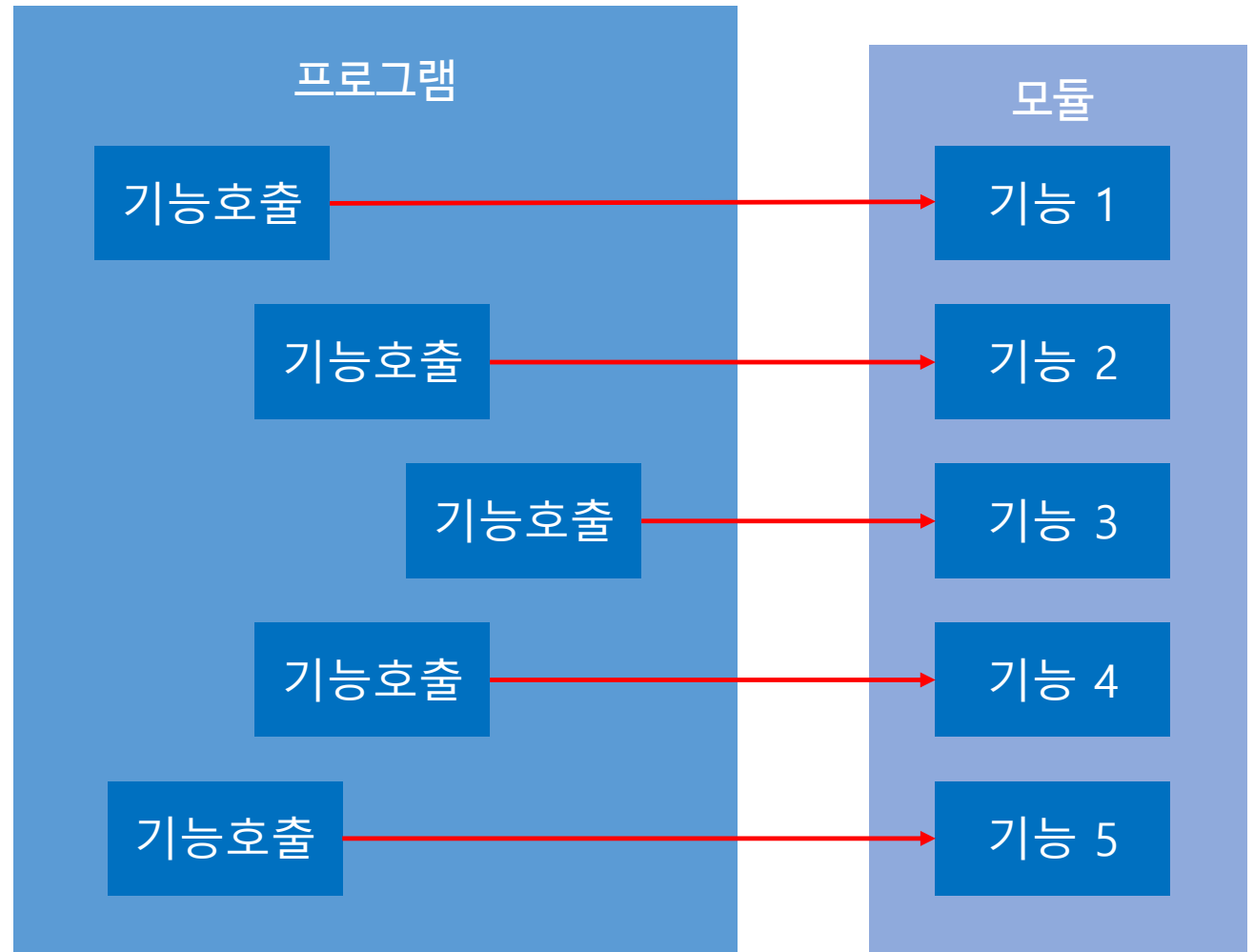
```

23-2 : 모듈이란?

일반적인 프로그램



모듈을 사용한 프로그램



23-3 : 직접만든 모듈

Ex : pjt_HelloWorld > jsFile02.js, jsFile03.js, jsFile04.js

모듈없이 기능구현

```

1 /**
2  * http://usejsdoc.org/
3  */
4
5  var sum = {};
6  sum.execution = function(num1, num2){
7      var result = 0;
8      for(var i=num1; i <= num2; i++){
9          result += i;
10     }
11
12     return result;
13 };
14
15 console.log(sum.execution(0, 10));
    
```

모듈을 직접 만들고 사용

exports

```

jsFile03.js
1 /**
2  * http://usejsdoc.org/
3  */
4
5  var sum = require("./sumNum");
6  console.log(sum.execution(0, 10));

sumNum.js
1 /**
2  * http://usejsdoc.org/
3  */
4
5  exports.execution = function(num1, num2){
6      var result = 0;
7      for(var i=num1; i <= num2; i++){
8          result += i;
9      }
10
11     return result;
12 };
    
```

module.exports

```

jsFile04.js
1 /**
2  * http://usejsdoc.org/
3  */
4
5  var sum = require("./sumNumObj");
6  console.log(sum.execution(0, 10));
7  console.log(sum.executionOdd(0, 10));

sumNumObj.js
1 /**
2  * http://usejsdoc.org/
3  */
4
5  var sumNum = {};
6
7  sumNum.execution = function(num1, num2){
8      var result = 0;
9      for(var i=num1; i <= num2; i++){
10         result += i;
11     }
12
13     return result;
14 };
15
16 sumNum.executionOdd = function(num1, num2){
17     var result = 0;
18     for(var i=num1; i <= num2; i++){
19         if(i % 2 != 0)
20             result += i;
21     }
22
23     return result;
24 };
25
26 module.exports = sumNum;
    
```

23-4 : 내부모듈

Ex : pjt_HelloWorld > jsFile05.js

node.js에서 기본적으로 제공하는 모듈

```
jsFile04.js
1 /**
2  * http://usejsdoc.org/
3  */
4
5 var os = require("os");
6 console.log(os);
7
8 console.log("=====");
9
10 console.log("os.hostname() : " + os.hostname());
11 console.log("os.type() : " + os.type());
12 console.log("os.platform() : " + os.platform());
13 console.log("os.uptime() : " + os.uptime());
14 console.log("os.networkInterfaces() : " + os.networkInterfaces());
```



```
Console
Debug
Markdown View
GFM View
Problems
<terminated> pjt_HelloWorld-jsFile04.js [Node Application] Node.js Process
[
  hostname: [Function: getHostname],
  uptime: [Function: getUptime],
  freemem: [Function: getFreeMem],
  totalmem: [Function: getTotalMem],
  cpus: [Function: getCPUs],
  type: [Function: getOSType],
  release: [Function: getOSRelease],
  networkInterfaces: [Function: getInterfaceAddresses],
  homedir: [Function: getHomeDirectory],
  userInfo: [Function: getUserInfo],
  loadavg: [Function: loadavg],
  constants:
    { UV_UDP_REUSEADDR: 4,
      errno:
        { E2BIG: 7,
          ...
        }
    }
]

=====
os.hostname() : DESKTOP-316GS26
os.type() : Windows_NT
os.platform() : win32
os.uptime() : 531937.1857662
os.networkInterfaces() : [object Object]
```

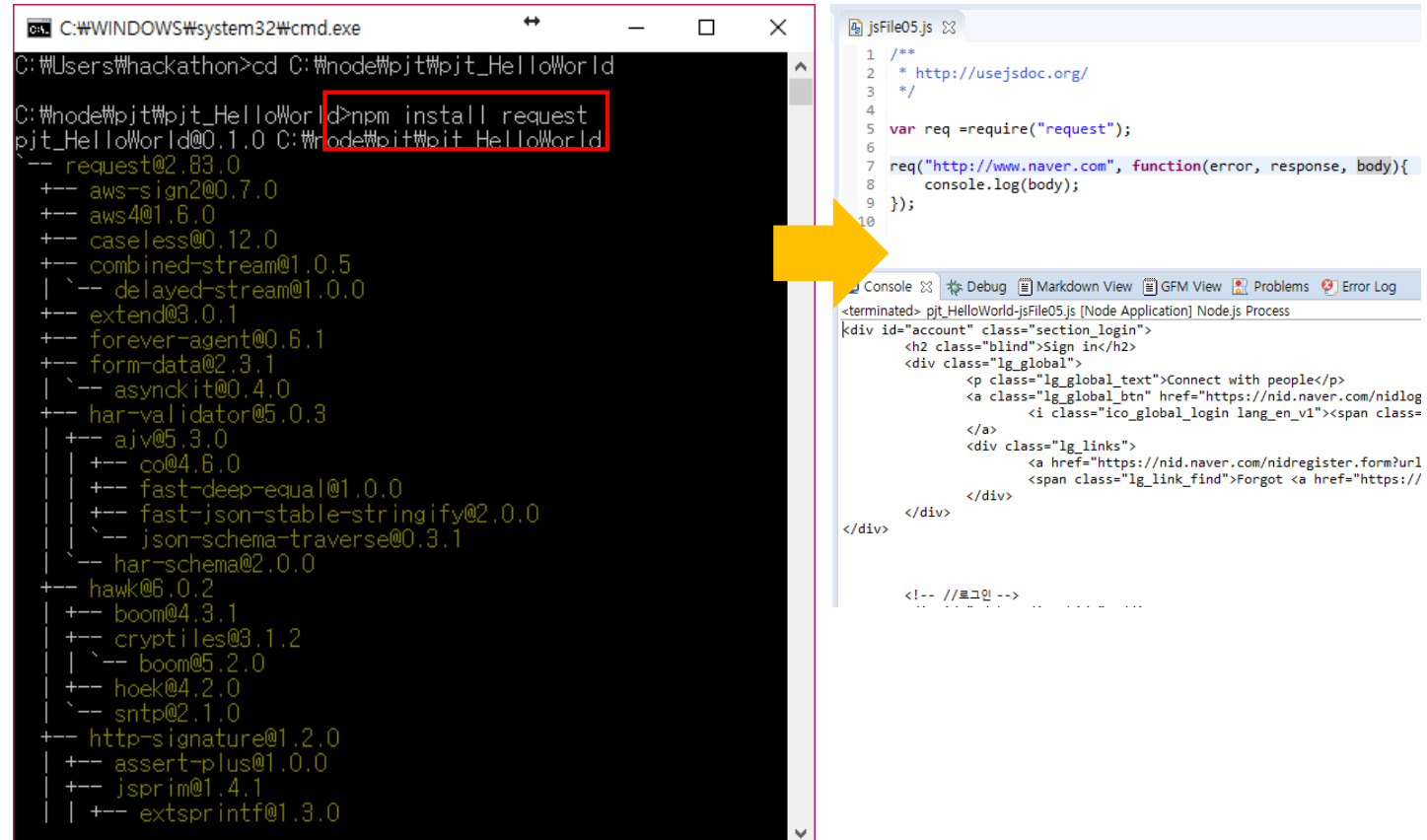
23-5 : 외부모듈

Ex : pjt_HelloWorld > jsFile05.js

node.js에서 제공하지 않는 모듈로 개인 또는 단체등에서 배포하고 <http://npmjs.org>에서 확인가능 함.

외부모듈 설치 방법
> npm install 외부모듈명

ex) request 모듈 설치
> npm install request



```
C:\WINDOWS\system32\cmd.exe
C:\Users\hackathon>cd C:\node\pjt\pjt_HelloWorld
C:\node\pjt\pjt_HelloWorld>npm install request
pjt_HelloWorld@0.1.0 C:\node\pjt\pjt_HelloWorld
-- request@2.83.0
+++ aws-sign2@0.7.0
+++ aws4@1.6.0
+++ caseless@0.12.0
+++ combined-stream@1.0.5
|   -- delayed-stream@1.0.0
+++ extend@3.0.1
+++ forever-agent@0.6.1
+++ form-data@2.3.1
|   -- async@2.6.0
+++ har-validator@5.0.3
|   ++ ajv@5.3.0
|   |   ++ co@4.6.0
|   |   ++ fast-deep-equal@1.0.0
|   |   ++ fast-json-stable-stringify@2.0.0
|   |   -- json-schema-traverse@0.3.1
|   -- har-schema@2.0.0
+++ hawk@6.0.2
|   ++ boom@4.3.1
|   ++ cryptiles@3.1.2
|   |   -- boom@5.2.0
|   ++ hoek@4.2.0
|   -- sntp@2.1.0
+++ http-signature@1.2.0
|   ++ assert-plus@1.0.0
|   ++ jsprim@1.4.1
|   |   ++ extsprintf@1.3.0
```

```
1 /**
2  * http://usejsdoc.org/
3  */
4
5  var req = require("request");
6
7  req("http://www.naver.com", function(error, response, body){
8    console.log(body);
9  });
10
```

```
<terminated> pjt_HelloWorld-jsFile05.js [Node Application] Node.js Process
<div id="account" class="section_login">
  <h2 class="blind">Sign in</h2>
  <div class="lg_global">
    <p class="lg_global_text">Connect with people</p>
    <a class="lg_global_btn" href="https://nid.naver.com/nidlog
      <i class="ico_global_login lang_en_v1"><span class=
    </a>
    <div class="lg_links">
      <a href="https://nid.naver.com/nidregister.form?url
        <span class="lg_link_find">Forgot <a href="https://
    </div>
  </div>
</div>

<!-- //크기인 -->
```