



실증 아카데미 리액트 특강

Node.js, npm, yarn의 기본



일시 2021년 01월 12일

주최 동아대학교





CONTENTS—

01 Node.js란?

JavaScript 런타임 환경
Node.js의 특징, 장단점
Node.js 설치

02 Visual Studio Code

텍스트 편집기
Visual Studio Code 설치
유용한 확장

03 브라우저와의 차이

04 Node.js 객체, 응용


Node.js 객체
process 객체 응용
fs, path 객체 응용

05 npm이란?

Node.js 패키지 매니저
npm 명령어
외부 모듈 install

06 yarn이란?

비공식 패키지 매니저
yarn 설치
yarn 명령어



01

Node.js란?

JavaScript 런타임 환경

Node.js의 특징, 장단점

Node.js 설치

JavaScript 런타임 환경

Node.js는 JavaScript 엔진인 크롬 V8에 libuv를 결합하여 빌드한 런타임 환경(플랫폼)

- JavaScript를 브라우저가 아닌 환경에서 실행할 수 있게 해주는 환경
- 프레임워크라고 알려져 있으나, 실제로는 C언어를 실행시켜주는 Visual Studio(gcc), Java를 실행시켜주는 Eclipse(JVM, JRE)와 같은 역할
- 보통 서버 언어로 알려져 있는 경우가 있으나, 이 또한 Node.js의 기능 중 하나이며, Express.js, Next.js라는 프레임워크가 서버 개발로 유명함
- 일반 브라우저와 같이 ECMAScript 사양을 따르나, 브라우저와는 환경이 다르므로 차이가 있을 수 있음



Node.js의 특징

- 고성능의 구글 V8 엔진 활용
- V8 엔진은 C++로 개발되어 확장성이 뛰어남
- 모듈 방식으로 소스 단위를 구성하며, 매우 방대한 오픈 소스 모듈을 제공
- 개발이 빠르고 쉬움

● 비동기 이벤트 기반

고성능의 비동기 방식으로, 한 가지 작업이 끝날 때까지 기다리지 않고 다음 작업을 처리하며, 작업이 끝나면 이벤트를 발생시켜 콜백 함수를 실행하는 방식

● 싱글 쓰레드

Node.js의 개발자는 한 가지 쓰레드만 관리하면 되므로 자원을 관리하는데 굉장히 용이하나, 실제로 내부 구현은 멀티 쓰레드로 되어 성능 또한 우수함

● 오류 처리

Node.js는 기본적으로 오류 핸들링(에러 처리)를 하지 않으면 예외 발생으로 애플리케이션이 종료됨
따라서 확실한 오류 핸들링이 필수



빠른 생산에 유리

어울리는 서비스

- 간단한 로직
- 빠른 응답시간, 빠른 개발 속도
- 비동기 방식에 어울리는 서비스(스트리밍, 채팅)

어울리지 않는 서비스

- 한 가지 처리가 오래 걸리는 경우
- 체크하는 로직이 많은 경우
- 업무 복잡도/난이도가 높은 경우



큰 규모에 적합
하지 않음

Node.js의 장점

01 JavaScript 기반

JavaScript 기반으로 Front-End 개발자가 Back-End 개발까지 쉽게 할 수 있어 높은 생산성을 보여줌

03 방대한 모듈

npm, yarn, github와 같은 패키지 매니저와 오픈 소스 저장소에 매우 방대한 모듈이 있기 때문에 필요한 모듈을 다운 받아 생산성 향상에 매우 큰 역할

02 고성능의 V8 엔진

구글에서 개발하는 V8 엔진은 매우 고성능으로, 느리다는 인식의 JavaScript를 매우 빨리 구동할 수 있음

또한, C++로 개발되어 확장성이 매우 좋음

04 비동기 방식

이벤트 기반 비동기 방식이므로 서버의 무리가 매우 적고, 내부적으로는 멀티 쓰레드이지만, 개발자는 싱글 쓰레드로 관리하므로 편리함



Node.js의 단점

01 작은 규모에 적합

CPU 부하가 큰 작업이나 대용량 파일을 관리하는 등 많은 I/O 등이 발생하면 성능 저하가 눈에 띄게 늘어나며, 오류 처리에 있어 매우 불안정

03 다른 설계 방식

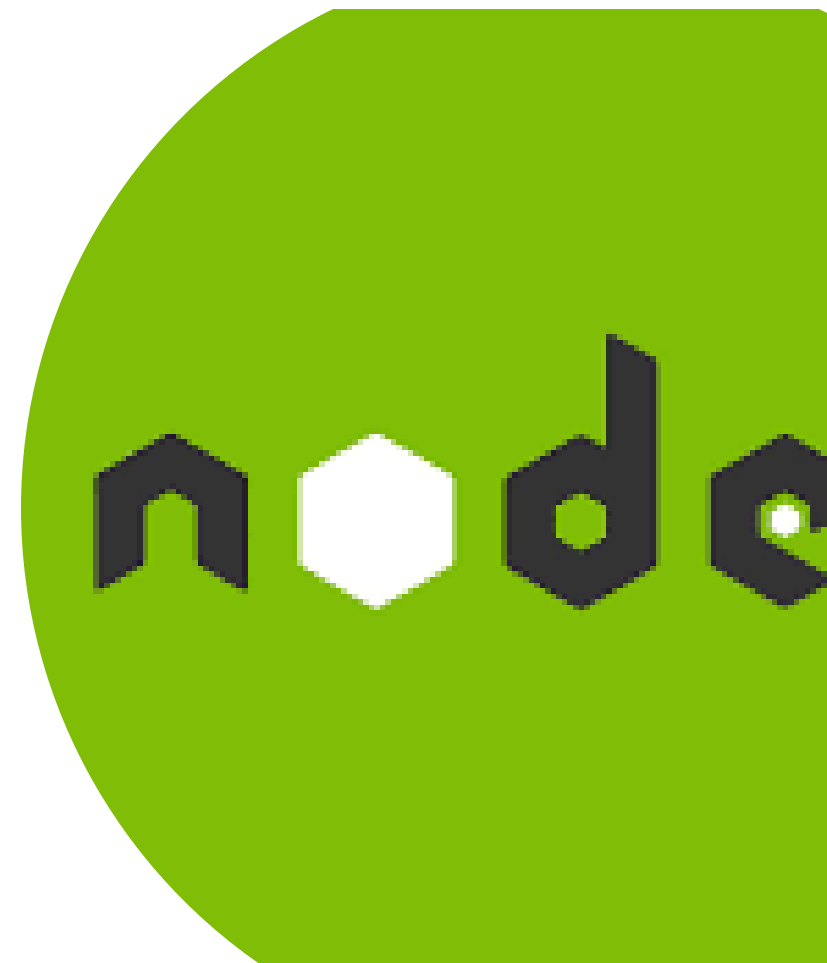
코드를 순차적으로 실행하는 것이 아니라 이벤트를 보내고 응답 이벤트가 오면 처리하는 방식이기 때문에 일반적인 프로그래밍과 같이 설계하면 어려움이 있을 수 있음

02 콜백함수

이벤트 기반 비동기 방식은 양날의 검으로, 로직이 복잡한 경우 콜백 함수의 늪에 빠져 소스코드가 매우 지저분해질 수 있음

04 에러 확인


JavaScript이기 때문에 코드가 실제로 실행되어야 에러가 있는지 알 수 있으며, 에러가 나면 애플리케이션이 종료되므로 테스트가 매우 중요



Node.js 설치

Node.js 홈페이지: <https://nodejs.org/ko/download/>

- 최신 버전은 기능이 불안정하거나 일부 모듈이 작동하지 않을 수 있으므로 LTS 버전 권장
- Node.js가 여러 버전이 있어 NVM이라는 버전 매니저를 통해 쉽게 다른 버전의 Node.js를 설치하고 전환할 수 있음
- 설치가 완료되고 터미널/CMD에서 `node -v`, `npm -v` 를 입력하면 버전 정보가 나옴 (나오지 않으면 재설치)



홈 | ABOUT | 다운로드 | 문서 | 참여하기 | 보안 | 뉴스 | CERTIFICATION


다운로드


최신 LTS 버전: **16.13.1** (includes npm 8.1.2)


플랫폼에 맞게 미리 빌드된 Node.js 인스톨러나 소스코드를 다운받아서 바로 개발을 시작하세요.

LTS
대다수 사용자에게 추천

현재 버전
최신 기능


Windows Installer
node-v16.13.1-x64.msi

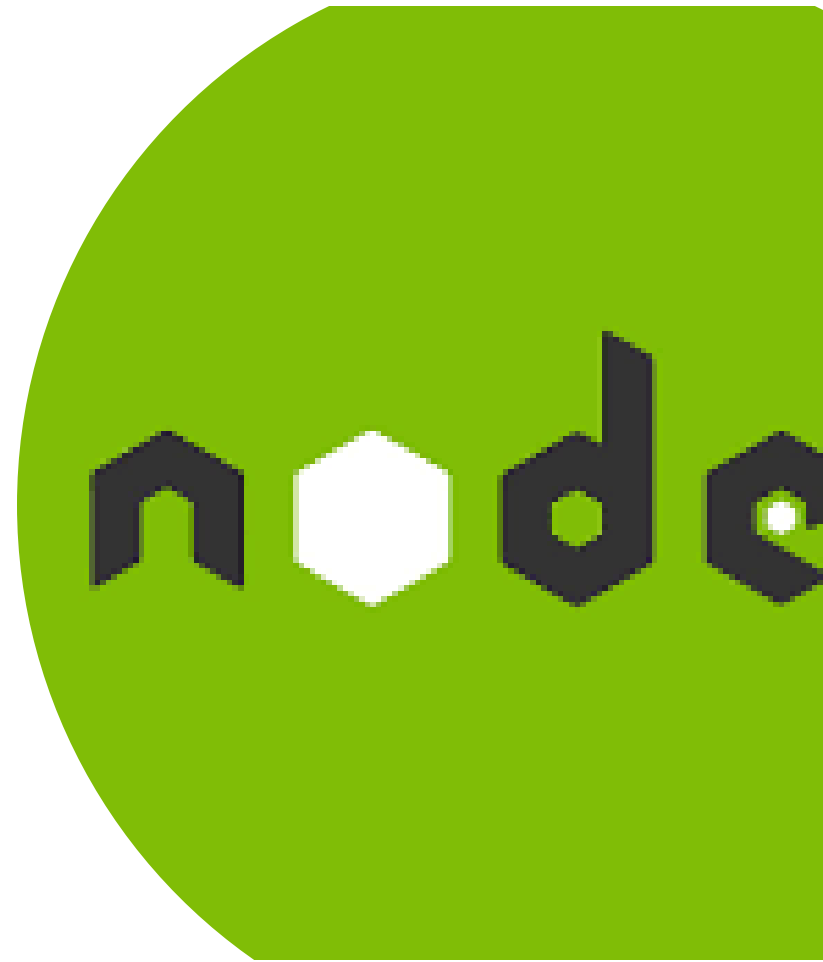

macOS Installer
node-v16.13.1.pkg


Source Code
node-v16.13.1.tar.gz

Windows Installer (.msi)
Windows Binary (.zip)
macOS Installer (.pkg)
macOS Binary (.tar.gz)
Linux Binaries (x64)
Linux Binaries (ARM)
Source Code

32-bit	64-bit
32-bit	64-bit
64-bit / ARM64	
64-bit	ARM64
64-bit	
ARMv7	ARMv8
node-v16.13.1.tar.gz	

그 밖의 플랫폼



macOS에서 Homebrew를 이용한 설치

1. 터미널에서 Homebrew 설치 명령어 입력

```
> /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

2. 설치 후 Homebrew를 통해 Node.js 설치

```
> brew install node
```



ubuntu(linux)에서 설치

1. 터미널에서 Node.js 설치 파일 다운로드

```
> curl -fsSL https://deb.nodesource.com/setup_16.x | sudo -E bash -
```

2. 관리자 권한으로 apt-get 패키지 관리자 실행하여 Node.js 설치

```
> sudo apt-get install -y nodejs
```





02

Visual Studio Code

텍스트 편집기

Visual Studio Code 설치

유용한 확장

텍스트 에디터

무거운 IDE와는 다르게 텍스트 파일을 편집하는 용도로 사용하는 가벼운 애플리케이션

- 주로 소프트웨어 개발에 사용
- 단순 텍스트 편집에서 환경 설정을 통해 IDE까지 넘보는 종류가 많음
- IDE는 개발의 모든 작업(코드 작성, 컴파일, 디버그, 소프트웨어 패키징, 배포) 등의 행동이 모두 포함되어 있어 무거우나, 텍스트 에디터는 코드 작성만을 위한 도구이기 때문에 가벼움
- JavaScript, Node.js는 각각 실행환경(컴파일, 디버그)이 브라우저, 명령어(터미널, CMD)이기 때문에 IDE까지는 필요 없음
- 메모장, Notepad++, Visual Studio Code, Atom, Sublime Text, vim, emacs 등이 있음



```
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146  
2147  
2148  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2156  
2157  
2158  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2210  
2211  
2212  
2213  
2214  
2215  
2216  
2217  
2218  
2219  
2220  
2221  
2222  
2223  
2224  
2225  
2226  
2227  
2228  
2229  
2230  
2231  
2232  
2233  
2234  
2235  
2236  
2237  
2238  
2239  
2240  
2241  
2242  
2243  
2244  
2245  
2246  
2247  
2248  
2249  
2250  
2251  
2252  
2253  
2254  
2255  
2256  
2257  
2258  
2259  
2260  
2261  
2262  
2263  
2264  
2265  
2266  
2267  
2268  
2269  
2270  
2271  
2272  
2273  
2274  
2275  
2276  
2277  
2278  
2279  
2280  
2281  
2282  
2283  
2284  
2285  
2286  
2287  
2288  
2289  
2290  
2291  
2292  
2293  
2294  
2295  
2296  
2297  
2298  
2299  
2300  
2301  
2302  
2303  
2304  
2305  
2306  
2307  
2308  
2309  
2310  
2311  
2312  
2313  
2314  
2315  
2316  
2317  
2318  
2319  
2320  
2321  
2322  
2323  
2324  
2325  
2326  
2327  
2328  
2329  
2330  
2331  
2332  
2333  
2334  
2335  
2336  
2337  
2338  
2339  
2340  
2341  
2342  
2343  
2344  
2345  
2346  
2347  
2348  
2349  
2350  
2351  
2352  
2353  
2354  
2355  
2356  
2357  
2358  
2359  
2360  
2361  
2362  
2363  
2364  
2365  
2366  
2367  
2368  
2369  
2370  
2371  
2372  
2373  
2374  
2375  
2376  
2377  
2378  
2379  
2380  
2381  
2382  
2383  
2384  
2385
```

Visual Studio Code

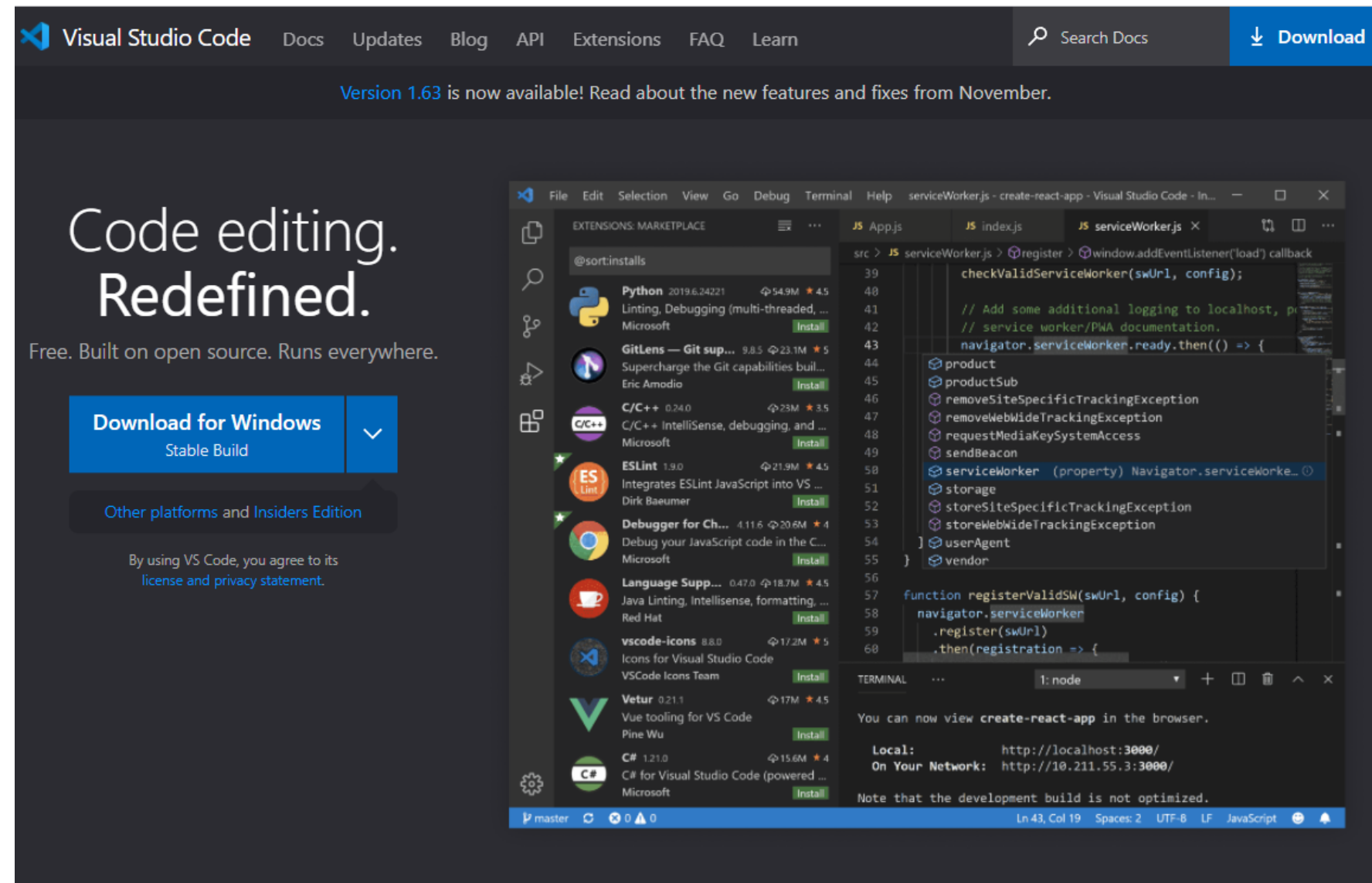
Microsoft에서 개발한 크로스 플랫폼 텍스트 에디터

- JavaScript, Node.js의 프레임워크 중 하나인 Electron.js로 개발
- 텍스트 에디터 내에 터미널/CMD가 내장되어 편리하게 명령어 사용
- IDE가 아니지만 컴파일 환경을 구축하면 C, C++ 등을 컴파일, 디버그 할 수 있음
즉, 노력 여하에 따라 IDE(Visual Studio, Eclipse)와 같은 동작 가능
- 강력한 플러그인, 확장 시스템으로 현재 가장 많이 사용되는 개발 도구
- JavaScript와 그 슈퍼셋 버전인 TypeScript와 궁합이 잘 맞음
- 다른 IDE와는 다르게 Windows, macOS 모두 사용 가능함




Visual Studio Code 설치

Visual Studio Code 홈페이지: <https://code.visualstudio.com/>




IntelliSense

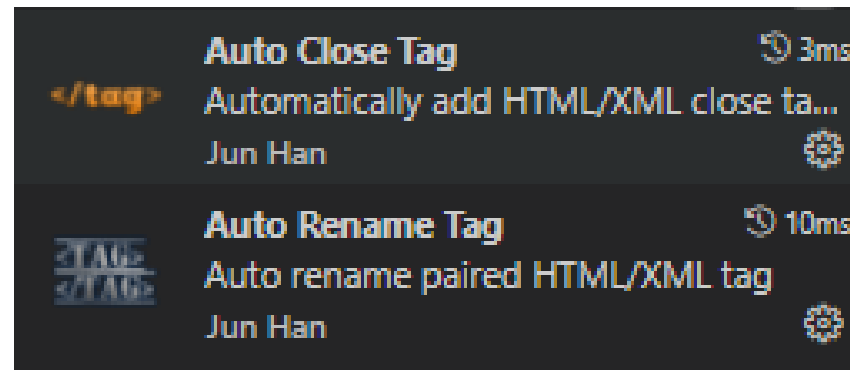

Run and Debug


Built-in Git

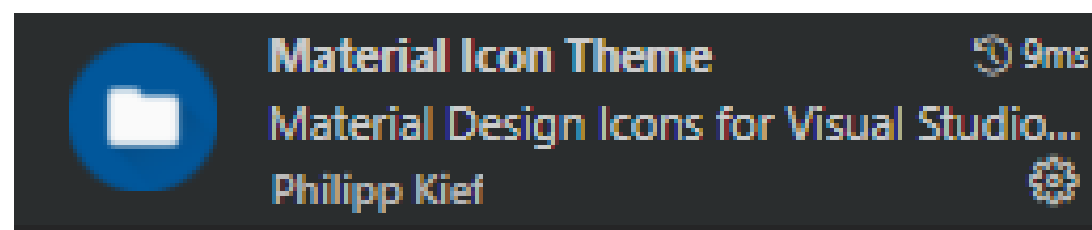

Extensions



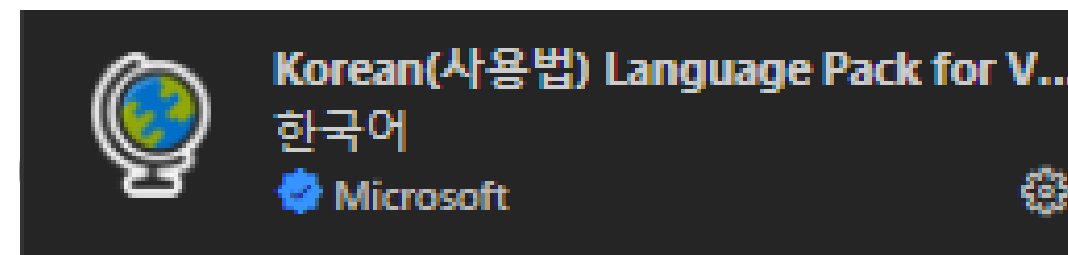
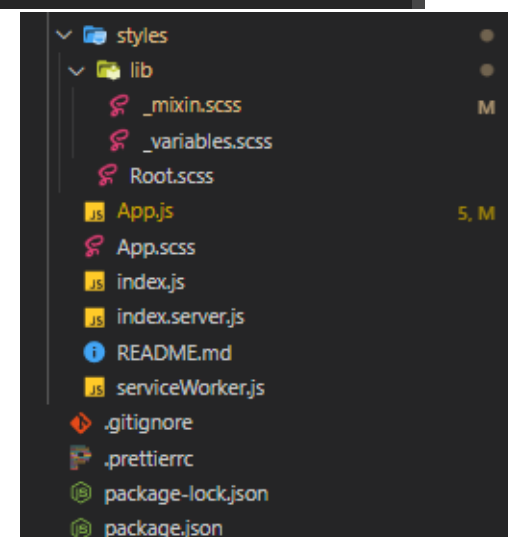
유용한 확장



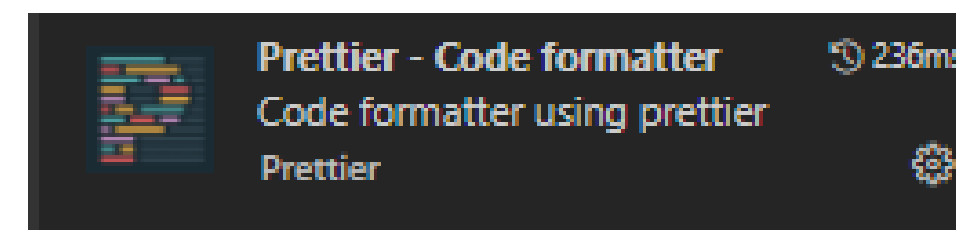
- Auto Close Tag, Auto Rename Tag
- HTML의 태그를 자동 완성해주고, 이름 변경 시 양 옆의 태그를 동시에 변경해주는 기능



- Material Icon theme
- 미관적 요소, 아이콘과 폴더를 쉽게 구분할 수 있게 해줌



- Korean Language Pack
- 설치 후 vscode를 재실행하면 한글 패치 완료



- Prettier - Code formatter
- 코드를 특정한 양식에 자동으로 맞춰주는 기능
소스코드에 양식을 두어 가독성을 높임





03

브라우저와의 차이

Node.js와 브라우저



- 개발 환경을 제공하는 것이 목적
- Node.js는 V8 엔진을 사용하고 있음
- Node.js는 버전을 선택할 수 있음
- Node.js는 CommonJS의 모듈 시스템을 사용함.
require를 통해 모듈 불러옴

Node.js API: <https://nodejs.org/docs/latest/api/>



- 웹 페이지를 화면에 띄워주는 것이 목적
- 브라우저마다 각기 다른 엔진을 사용하고 있음
(Chrome의 경우는 V8 엔진)
- 브라우저는 환경을 선택할 수 없음
JavaScript의 발전 속도를 브라우저가 따라가지 못 하기도 함
- 브라우저는 ES 모듈 표준을 사용하고 있음
import로 모듈 불러옴

브라우저 API: <https://developer.mozilla.org/ko/docs/Web/API>

04

Node.js 객체, 응용

Node.js 객체
process 객체 응용
fs, path 객체 응용

Node.js의 객체

- Node.js는 프로세스 단계에서 실행되기 때문에, 다른 언어와 마찬가지로 파일 입출력, 경로, 현재 실행 중인 컴퓨터의 정보 등을 가져올 수 있음
- 이러한 기능을 수행하는 고성능의 빌트인(내장) 객체 제공
- 내장 객체의 정보를 알고, 잘 사용하는 것이 Node.js 개발의 시작

※ 브라우저에서 사용하는 DOM 객체(document, window 등)은 사용할 수 없음

- **process 객체**

프로세스를 관리하는 객체

시스템 정보와 현재 프로세스의 상태 등을 제어하는 역할

- **global, globalThis 객체**

Node.js에서 전역 객체로, 브라우저의 window 객체와 비슷한 역할을 수행

어디서든 공통 객체 접근 가능

- **fs, os, path 등의 기본 제공 모듈**

fs모듈은 파일 입출력을 담당

os모듈은 process와 비슷하지만 운영체제와 시스템 정보, 컴퓨터 정보를 가져올 수 있음

path 모듈은 파일/폴더의 경로를 혼란 없이 사용할 수 있게 해줌

process 객체

프로세스를 관리하는 객체

- 기본 사용법 예제

```
console.log(process.env);    // 컴퓨터 환경과 관련된 정보를 가진 객체
console.log(process.version); // node.js의 버전
console.log(process.versions); // node.js와 연관된 프로그램들의 버전을 가진 객체
console.log(process.arch);    // 프로세서의 아키텍처(arm/ia32/x64)
console.log(process.platform); // 플랫폼(win32/linux/sunos/freebsd/darwin)
console.log(process.memoryUsage()); // 메모리 사용 정보를 가진 객체
console.log(process.uptime()); // 현재 프로그램이 실행된 시간
```

- 애플리케이션 종료

```
process.exit();
process.exit(0); // 정상 종료

process.exit(-1); // 비정상 종료
```

- 운영체제 구분

```
if (process.platform === "darwin") {
    // macOS일 때 실행할 코드
} else {
    // Windows일 때 실행할 코드
}
```



fs, path 모듈

경로를 혼란없이 사용하여 파일 입/출력 처리

- 기본 사용법(비동기) 예제

```
const fs = require("fs");
const path = require("path");
const file_path = path.join(__dirname, "test.txt");
fs.writeFile(file_path, new Date().toLocaleString(), (err) => {
  console.log("write Error", err);
  fs.readFile(
    file_path,
    {
      encoding: "utf8",
    },
    function (err, data) {
      console.log("read Error", err, data);
    }
  );
});
```

- 동기 사용 예제

```
const fs = require("fs");
const path = require("path");

const sync_file_path = path.join(__dirname, "sync_test.txt");
fs.writeFileSync(sync_file_path, new Date().toLocaleString());
const sync_read_file = fs.readFileSync(sync_file_path, {
  encoding: "utf8",
});
console.log("sync read file:", sync_read_file);
```



fs, path 모듈

- fs 모듈은 파일 입출력을 담당하며, Sync가 아닌 비동기 방식은 콜백 함수를 넘겨줘야 함
- 위 코드처럼 JavaScript는 화살표 함수라는 것이 존재하며, 이를 콜백 함수로 넘긴 예시
- ※ 화살표 함수는 추후 자세히 다룸
- JavaScript는 이런 비동기에 콜백 함수가 많이 들어가기 때문에 순서 보장을 위해 콜백 지옥이 발생
- 때문에 콜백을 최소화 하는 문법이 많이 생김

```
console.log("Pyramid of dooooooooooom!💣💣💣")

step1(function (value1) {
  step2(function (value2) {
    step3(function (value3) {
      step4(function (value4) {
        step5(function (value5) {
          step6(function (value6) {
            step7(function (value7) {
              //Do something with value 4
              console.log(value7)
            });
          });
        });
      });
    });
  });
});
```

- __dirname은 현재 폴더의 경로를 가져옴
- path 모듈의 join은 여러 문자열을 한 가지 경로로 합쳐주는 역할을 함
- 합쳐주면서 각 OS별 다른 경로 형식을 적용
- '../', '../..', './' 등의 폴더 이동 경로 형식도 적용됨
- new Date()는 현재 시간을 나타내는 Date 객체



05

npm이란?

Node.js 패키지 매니저

npm 명령어

외부 모듈 install

npm이란? Node.js 공식 패키지 매니저

npm 홈페이지: <https://www.npmjs.com/>



JavaScript 모듈
방대하게 흩어짐



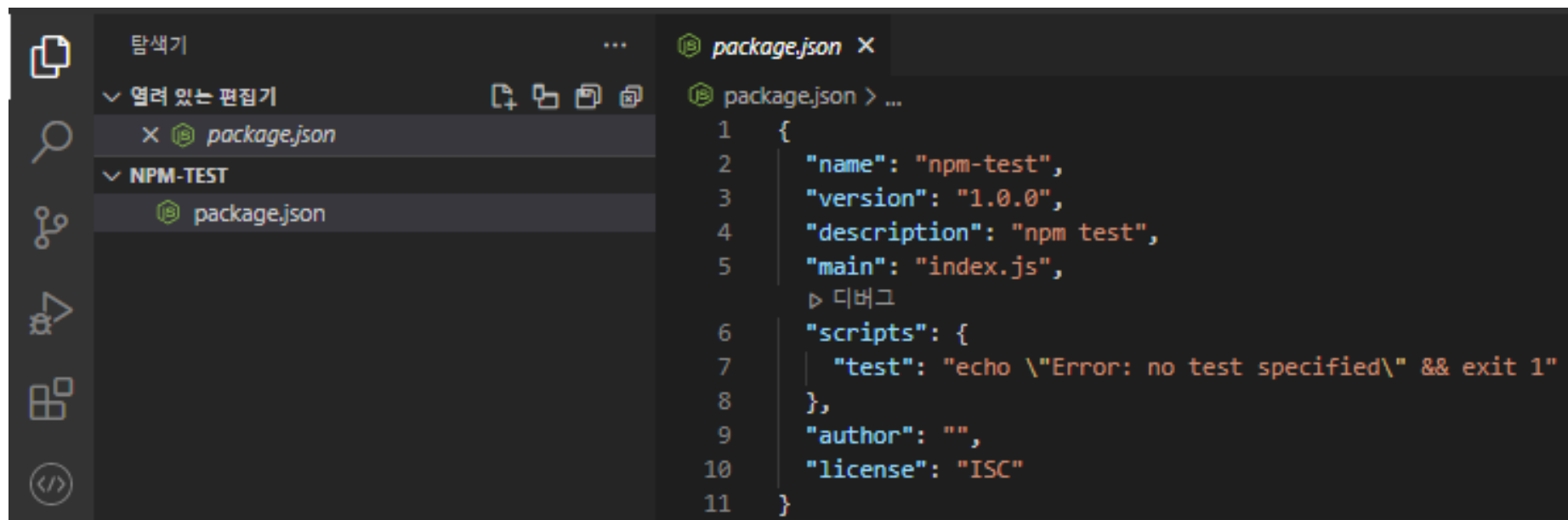
- 기존의 모듈은 방대하게 흩어져 있음
- github에서 직접 clone하는 불편한 방식
- 찾기 매우 어려운 상태

- Node Package Manager(Module)
- Node.js 패키지를 관리할 수 있는 도구, 저장소
- 140만개 이상의 패키지 등록

npm 설정 명령어

npm init

1. 터미널/CMD를 켜고 후 `cd Desktop` 명령어를 통해 바탕화면 진입
2. `mkdir npm-test` 명령어를 통해 npm-test라는 폴더 생성
3. `cd npm-test` 명령어를 입력하면 npm-test 폴더로 진입
4. `npm init` 명령어를 입력한 후 여러 가지 문항이 나오는데, 모두 Enter를 입력하면 해당 폴더에 package.json이 생성
5. `code .` 명령어를 입력하여 vscode를 실행



package.json과 /node_modules

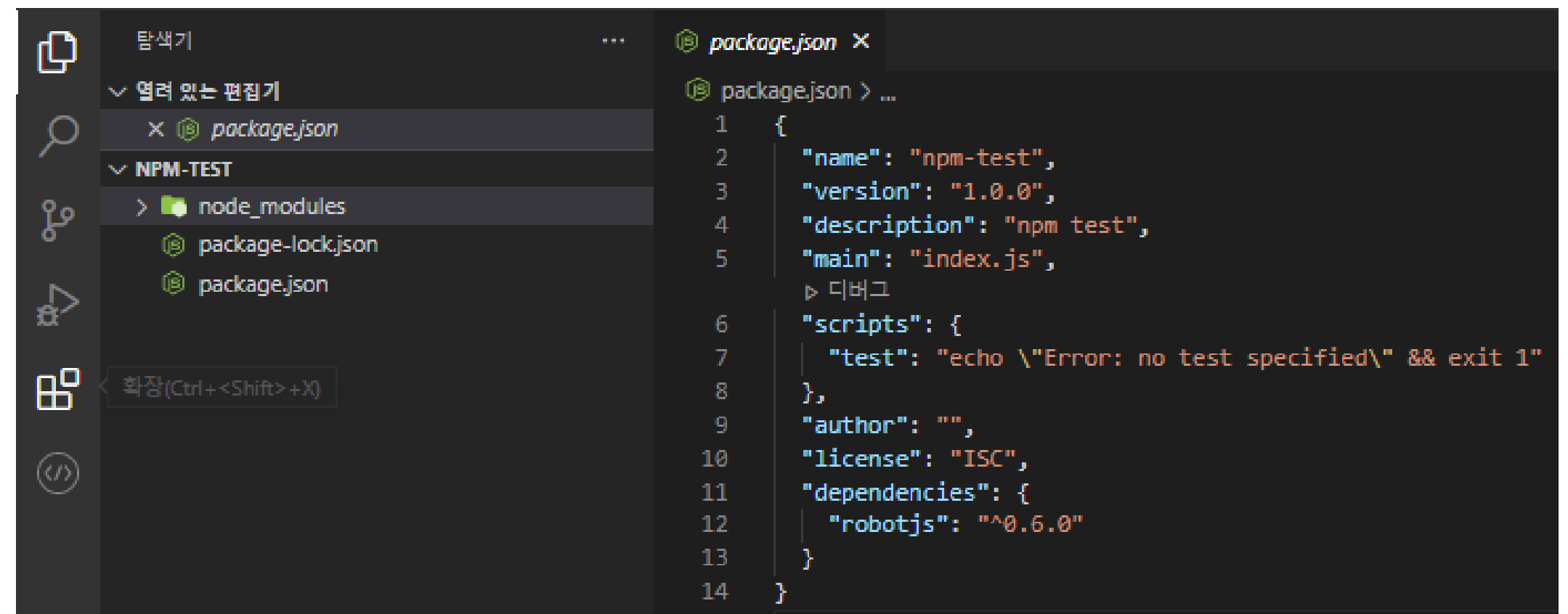
npm 프로젝트에 대해 설정하는 명세서와 의존 모듈 저장 폴더

package.json의 내용

- "name": 프로젝트의 이름
- "version": 프로젝트의 버전
- "description": 프로젝트의 설명
- "scripts": 커스텀 npm 명령어를 정의
- "dependencies": 의존 중인 모듈을 명시
- "devDependencies": 개발할 때만 사용하는 의존 모듈 명시

/node_modules

- 의존 중인 모듈을 실제로 저장하는 로컬 저장소
- npm install 명령어 실행 시 package.json에 명시된 의존 모듈을 모두 설치함
- 모듈이 많이 설치될 수록 용량이 커지며, 폴더가 손상되거나 삭제되더라도 package.json만 있다면 복구 가능
- 삭제된 상태에서 의존 모듈을 사용하려 할 경우 not found 에러 발생



npm install, uninstall

npm 프로젝트에 모듈을 추가/제거하는 명령어

- npm install 명령어를 단일로 입력할 경우, package.json에 명시된 모듈 모두 설치
- npm install <모듈 명> 명령어를 통해 현재 프로젝트에 dependencies로 해당 모듈 추가
- npm install <모듈 명> --save-dev 명령어는 devDependencies로 추가
- npm uninstall <모듈 명> 명령어를 통해 현재 프로젝트에서 모듈 제거
- npm 홈페이지에서 모듈 검색 가능
- 일반적으로 모듈은 설명하는 문서가 해당 모듈의 홈페이지에 기술되어 있음

> npm install robotjs 명령어를 통해 robotjs 모듈을 추가하여 보세요.

※ robotjs 모듈은 python 2.7 버전이 설치 되어야 하며,

Windows의 경우 Visual Studio 2013 이상, macOS의 경우 XCode가 추가로 설치되어야 합니다.



npm run, npm start

npm 커스텀 명령어를 실행하는 명령어

- 프로젝트에 index.js 파일 생성

```
// Move the mouse across the screen as a sine wave.
```

```
const robot = require("robotjs");
```

```
// Speed up the mouse.
```

```
robot.setMouseDelay(1);
```

```
const twoPI = Math.PI * 2.0;
```

```
const screenSize = robot.getScreenSize();
```

```
const height = screenSize.height / 2 - 10;
```

```
const width = screenSize.width;
```

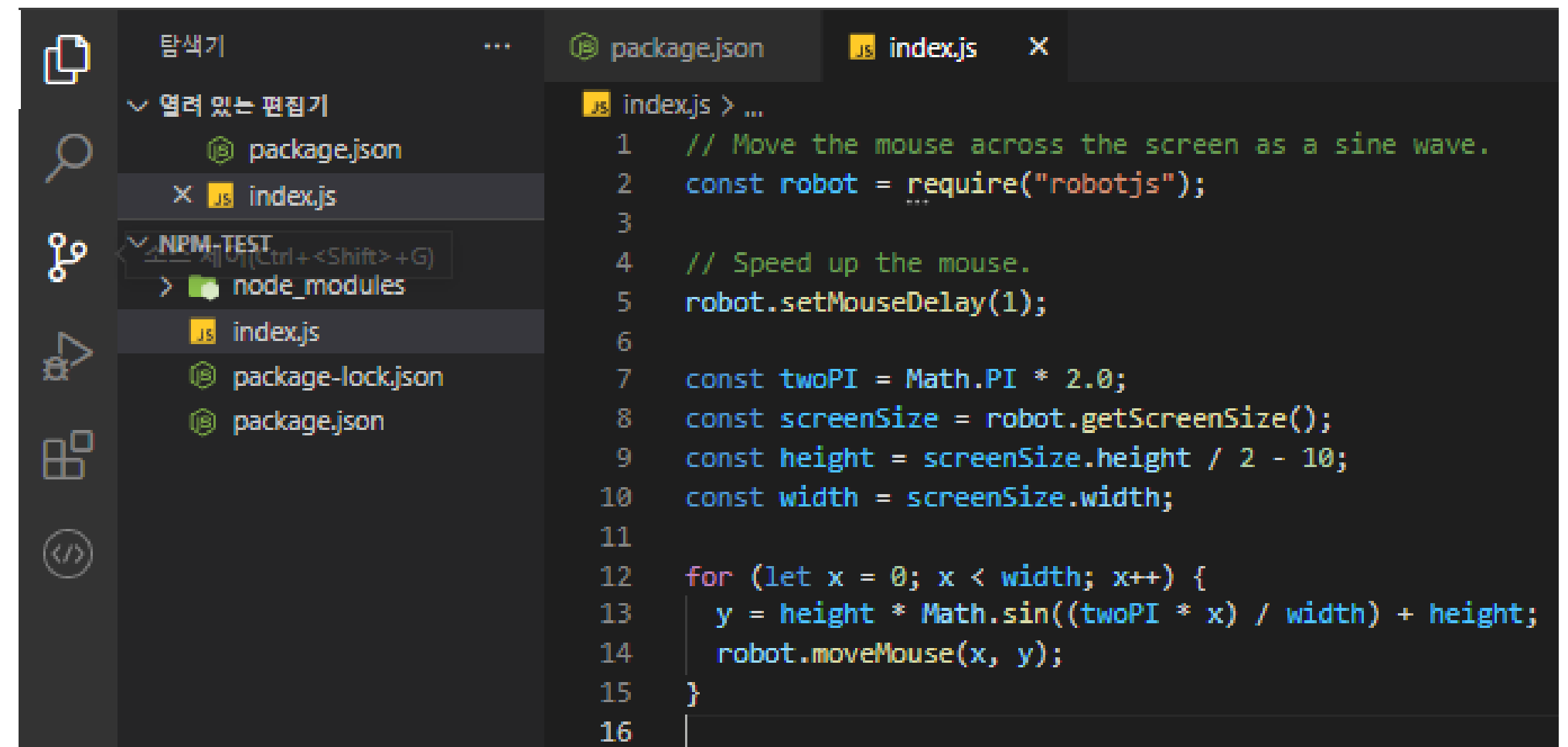
```
for (let x = 0; x < width; x++) {
```

```
  y = height * Math.sin((twoPI * x) / width) + height;
```

```
  robot.moveMouse(x, y);
```

```
}
```

- index.js 파일에 해당 코드를 작성



The screenshot shows a code editor with a dark theme. On the left, the file explorer shows a project named 'NPM-TEST' with a 'node_modules' folder and several files including 'index.js'. The main editor area shows the content of 'index.js' with the following code:

```
1 // Move the mouse across the screen as a sine wave.
2 const robot = require("robotjs");
3
4 // Speed up the mouse.
5 robot.setMouseDelay(1);
6
7 const twoPI = Math.PI * 2.0;
8 const screenSize = robot.getScreenSize();
9 const height = screenSize.height / 2 - 10;
10 const width = screenSize.width;
11
12 for (let x = 0; x < width; x++) {
13   y = height * Math.sin((twoPI * x) / width) + height;
14   robot.moveMouse(x, y);
15 }
16
```

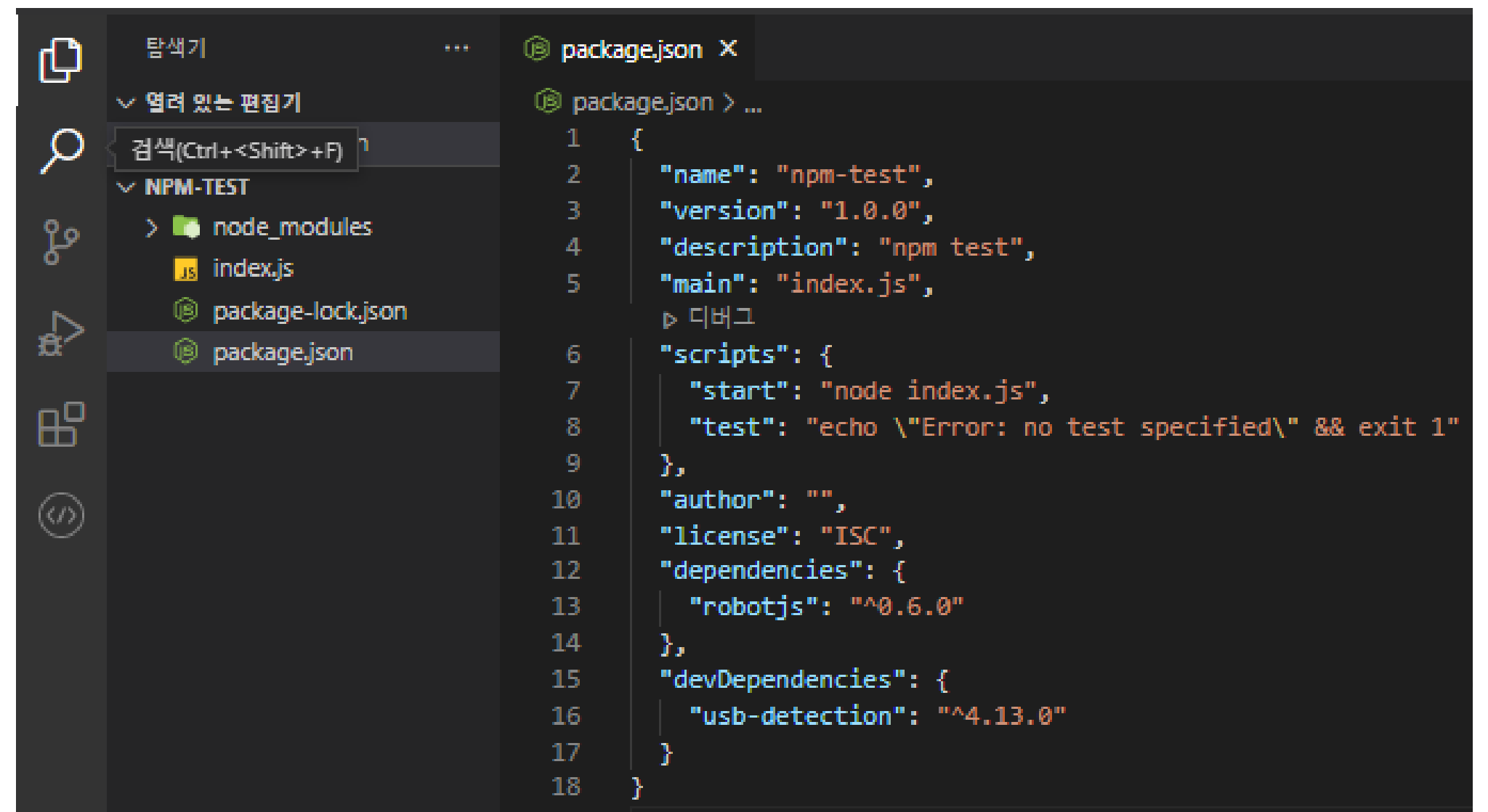
npm run, npm start

npm 커스텀 명령어를 실행하는 명령어

- package.json의 scripts에 start 작성

```
"scripts": {  
  "start": "node index.js",  
  "test": "echo W\"Error: no test specifiedW\" && exit 1"  
},
```

- node 명령어는 Node.js 파일을 실행하는 명령어
- npm run 명령어는 scripts의 커스텀 명령어를 실행
- npm run start를 실행하면 node index.js 실행
- start 명령의 경우 특별히 npm start로도 사용 가능



The screenshot shows a code editor with two panels. The left panel displays a file explorer for a project named 'NPM-TEST'. It shows a 'node_modules' directory, an 'index.js' file, and two JSON files: 'package-lock.json' and 'package.json'. The 'package.json' file is selected. The right panel shows the content of 'package.json', which is a JSON object with the following fields: 'name' (npm-test), 'version' (1.0.0), 'description' (npm test), 'main' (index.js), 'scripts' (start: node index.js, test: echo \"Error: no test specified\" && exit 1), 'author' (empty), 'license' (ISC), 'dependencies' (robotjs: ^0.6.0), and 'devDependencies' (usb-detection: ^4.13.0).

```
package.json X  
package.json > ...  
1  {  
2    "name": "npm-test",  
3    "version": "1.0.0",  
4    "description": "npm test",  
5    "main": "index.js",  
6    "scripts": {  
7      "start": "node index.js",  
8      "test": "echo \"Error: no test specified\" && exit 1"  
9    },  
10   "author": "",  
11   "license": "ISC",  
12   "dependencies": {  
13     "robotjs": "^0.6.0"  
14   },  
15   "devDependencies": {  
16     "usb-detection": "^4.13.0"  
17   }  
18 }
```

06

yarn이란?

비공식 패키지 매니저

yarn 설치

yarn 명령어

yarn이란?

페이스북에서 발표한 비공식 패키지 매니저

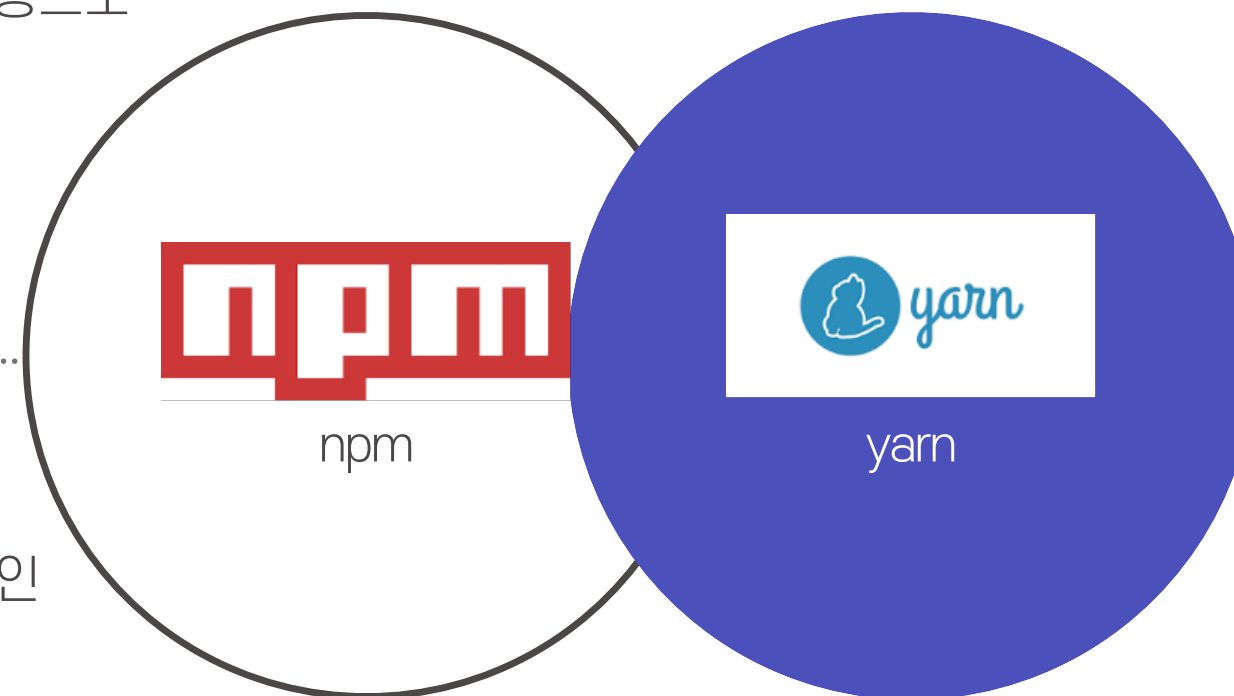
yarn 홈페이지: <https://yarnpkg.com/>

초창기에는 npm을 압도하는 성능으로 인기를 끌었으나 현재는 npm이 개선을 거듭하여 큰 차이가 없음

npm과 동일한 저장소를 사용하므로 yarn import를 통해 손쉽게 마이그레이션

npm은 package-lock.json, yarn은 yarn.lock 파일로 구체적인 의존성 관리, yarn 패키지 알고리즘 상 npm과는 달리 모든 환경에서 동일한 의존성 보장

현재 node_modules의 근본적인 문제점을 해결하기 위해 .pnp.js라는 새로운 방식의 통합 의존성 파일 모색



npm과 yarn 비교: <https://dongmin-jang.medium.com/npm-vs-yarn-7b699c5d6034>

yarn 설치

1. 터미널/CMD에서 npm을 통해 yarn 설치 명령어 입력

```
> npm install -g yarn
```

혹은

```
> npm i -g yarn
```

2. 설치 후 터미널/CMD에서 버전 확인

```
> yarn --version
```

※ npm install에서 -g 옵션은 npm 모듈을 전역적으로 설치하겠다는 뜻

즉, PC 어디에서도 해당 모듈에 접근 가능하게 되며, yarn은 npm 모듈로써 명령어를 제공하고 있음

전역 설치를 하고 싶지 않다면 npm install yarn을 사용한 후에 커스텀 명령어로 해당 프로젝트에서만 사용 가능



yarn 명령어

- yarn init: npm init과 동일, package.json이 있으면 실행하지 않아도 됨
- yarn: npm install과 동일. package.json에 명시된 모듈들을 설치
- yarn add <모듈 명>: npm install <모듈 명>과 동일
- yarn remove <모듈 명>: npm uninstall <모듈 명>과 동일
- yarn add -D <모듈 명>: npm install --save-dev <모듈 명>과 동일
- yarn <커스텀 명령어>: npm run <커스텀 명령어>와 동일



실증 아카데미 리액트 특강

THANK
YOU

(주) 인바이즈 박철현
(주) 인바이즈 개발팀