|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **영어** | **뜻** | **No.** | **영어** | **뜻** |
| 1. | add / remove | 추가 / 삭제 | 46. | stop  end  finish | 재시작 가능한 멈춤  재시작 불가능한 멈춤  완료했다는 의미 |
| 2. | start / stop | 시작 / 정지 |
| 3. | begin / end | 시작 / 끝 |
| 4. | first / last | 처음 / 마지막 | 47. | stop  suspend  pause | 움직임의 멈춤  일시정지  재개할 수 있는 멈춤 |
| 5. | top / last | 선두 / 말미 |
| 6. | head / tail | 머리 / 꼬리 |
| 7. | header / footer | 헤더 / 푸터 | 48. | quit  exit | 현재 상태에서 탈출  출구로 나가는 것 |
| 8. | top / bottom | 최상 / 최하 |
| 9. | up / down | 위 / 아래 | 49. | changed  modified  revised | 전체적인 변경  부분적인 수정, 개선  개정 |
| 10. | Upper case / Lower case | 대문자 / 소문자 |
| 11. | under / over | 미만 / 초과 |
| 12. | or under / and over | 이하 / 이상 | 50. | find / search | 찾다 / 검색하다 |
| 13. | greater than | ~보다 큰, >기호 | 51. | to☆/parse☆/convert☆ | ☆로 변환 |
| 14. | less than | ~보다 작은, < 기호 | 52. | tryParse☆ | ☆로 변환 시도 |
| 15. | previous / next | 앞으로 / 다음으로 | 53. | from☆ | ☆로부터 변환 |
| 16. | forward / backward | 앞으로 / 뒤로 | 54. | clear  delete | 파일의 내용을 지움  파일 자체를 지움 |
| 17. | foreground/background | 전경 / 배경 |
| 18. | push / pop | 스택 삽입 / 스택 삭제 | 55. | create  make  generate | 없던 걸 생성  무언가를 참고 후 생성  무언가를 변환 후 생성 |
| 19. | push / pull | Git 삽입 / Git 삭제 |
| 20. | enqueue / dequeue | 큐 삽입 / 큐 삭제 |
| 21. | before / after | 전 / 후 | 56. | parameter / argument | 매개변수 / 인자 |
| 22. | parent/child/children/self | 부모/자식/자식들/자신 | 57. | property / attribute | 속성 |
| 23. | open / close | 열기 / 닫기 | 58. | number / numeric | 숫자,번호 / 10진수숫자 |
| 24. | connect / disconnect | 네트워크 연결 / 해제 | 59. | sum / total | 합 / 총량 |
| 25. | input / output | 입력 / 출력 | 60. | limits  bounds  range | 제한  수학적 느낌의 제한선  값의 범위 |
| 26. | import / export | 가져오기 / 내보내기 |
| 27. | read / write | 읽기 / 쓰기 |
| 28. | save / load | 저장 / 불러오기 | 61. | top  peak  spike | 최상, 선두  쌓인 것 중 최상  꺾인 선 그래프의 최상 |
| 29. | set / unset | 설정 / 해제 |
| 30. | lock / unlock | 잠금 / 해제 |
| 31. | attach / detach | 붙이다 / 떼다 | 62. | exclude / ignore | 배제하다 / 무시하다 |
| 32. | suspend / resume | 쉬다 / 재개하다 | 63. | state / status | 상태 |
| 33. | enable / disable | 유효한 / 무효한 | 64. | letter / character | 문자 |
| 34. | activate / deactivate | 활성화 / 비활성화 | 65. | title / caption | 제목 |
| 35. | show / hide | 표시 / 숨김 | 66. | issue  problem | 논해야 하는 문제  해결이 필요한 문제 |
| 36. | visible / invisible | 보임 / 보이지 않음 |
| 37. | available / unavailable | 이용 가능/이용 불가능 | 67. | individual  personal  private | 여러 개체 중 하나  개인적인  사적인 |
| 38. | allow / deny | 허가 / 불허 |
| 39. | host / guest | 주인 / 손님 |
| 40. | server / client | 서버 / 클라이언트 | 68. | public | 공적인 |
| 41. | provider / user | 제공자 / 사용자 | 69. | just / only | 단지 / 오직 |
| 42. | resume / pause | 재개 / 일시정지 | 70. | within | 시간, 거리, 범위 이내 |
| 43. | create / destroy | 생성 / 파괴 | 71. | fix | 수리, 고정 |
| 44. | config/setting/preference | 구성 / 설정 / 환경설정 | 72. | apply | 적용하다 |
| 45. | initialize / init / setup | 초기화 | 73. | flush | 쌓인 데이터를 비우다. |
| **No.** | **영어** | **뜻** | **No.** | **영어** | **뜻** |
| 74. | validate, verify | 유효하게 하다 | 119. | contains | 포함 |
| 75. | syntax | 구문 | 120. | equals | 같음 |
| 76. | statement | 문장 (for문,if문,함수 등) | 121. | point / x / y / offset | 좌표/x축/y축/상대위치 |
| 77. | expression | 평가식 | 122. | size / width / height | 크기 / 폭 / 높이 |
| 78. | operator | 연산자 | 123. | top / middle / bottom | 상단 / 중단 / 하단 |
| 79. | signature | 타입의 표현 | 124. | left / center / right | 왼쪽 / 중앙 / 오른쪽 |
| 80. | inheritance | 상속 | 125. | front / middle / back | 앞 / 중간 / 뒤 |
| 81. | delimiter / separator | 구분자 | 126. | front / side / rear | 앞쪽 / 옆쪽 / 뒤쪽 |
| 82. | Log | 기록, 로그 | 127. | fill | 색칠하다 |
| 83. | stack | 스택, 쌓다 | 128. | depth | 깊이, 심도 |
| 84. | token | 토큰, 최소 단위 |  |  |  |
| 85. | optimize | 최적화하다 |  |  |  |
| 86. | normalize | 정규화하다 |  |  |  |
| 87. | cheatsheet | 사용법을 정리해놓은 것 |  |  |  |
| 88. | usage | 사용방법 |  |  |  |
| 89. | unknown | 미지의 |  |  |  |
| 90. | misc / miscellaneous | 다양한, 다방면의 |  |  |  |
| 91. | description | 설명 |  |  |  |
| 92. | brief | 개요, 요약하다 |  |  |  |
| 93. | features | 특징, 기능 |  |  |  |
| 94. | via | ~에 의해 |  |  |  |
| 95. | required | 필수의, 요구되는 |  |  |  |
| 96. | Initial(initialized) value | 초기치, 초기값 |  |  |  |
| 97. | default value | 기본값 |  |  |  |
| 98. | original value | 변경 전 값 |  |  |  |
| 99. | current value | 현재 값 |  |  |  |
| 100. | return value | 반환 값 |  |  |  |
| 101. | variable / var | 변수 |  |  |  |
| 102. | literal | 직접 기술한 값 |  |  |  |
| 103. | constant / const | 상수 |  |  |  |
| 104. | primitive data type | 기본형 데이터 타입 |  |  |  |
| 105. | null / nil | 존재하지 않는 것 |  |  |  |
| 106. | void | 빈 것 |  |  |  |
| 107. | prefix / suffix | 접두어 / 접미어 |  |  |  |
| 108. | release | 출시 |  |  |  |
| 109. | binary number | 2진수 |  |  |  |
| 110. | octal number | 8진수 |  |  |  |
| 111. | decimal number | 10진수 |  |  |  |
| 112. | hexadecimal number | 16진수 |  |  |  |
| 113. | odd / even | 홀수 / 짝수 |  |  |  |
| 114. | true / false | 참 / 거짓 |  |  |  |
| 115. | is☆ | ☆인가? |  |  |  |
| 116. | has☆ | ☆를 갖는가? |  |  |  |
| 117. | can☆ | ☆할 수 있는가? |  |  |  |
| 118. | exists | 존재 |  |  |  |

1. manual : 매뉴얼은 프로그램들의 사용과 운영에 대한 내용이 기술되어 있는 문서

2. agile model : 애자일 모형은 소프트웨어 개발과정에서 지속적으로 발생하는 변경에 유연하고 기민하게 대응하여 생산성과 품질 향상을 목표로 하는 협력적 소프트웨어 개발 방법론. 프로젝트의 생명주기 동안 개발 팀원들 간의 상호작용과 고객과의 협업을 중심으로 반복적, 점진적인 계획을 통해 요구사항의 변화를 관리하고 또한 문서 작업보다 코딩과 테스트 기반의 접근을 통해 소프트웨어를 개발하는 방식이다. 익스트림 프로그래밍(Extreme Programming, XP), 스크럼(Scrum), Lean 등이 있다.

3. backlog : 백로그는 제품 개발에 필요한 요구사항을 모두 모아 우선순위를 부여해 놓은 목록

4. release : 릴리즈는 몇 개의 요구사항이 적용되어 부분적으로 기능이 완료된 제품을 제공하는 것.

5. visibility : 가시성이란 대상을 확인 할 수 있는 정도를 의미. 릴리즈 기간을 짧게 반복하면서 개발 과정에서 제품 소프트웨어의 일부 기능이 구현될 때마다 고객에게 이를 확인시켜주면, 고객은 요구사항이 잘 반영되고 있음을 직접적으로 알 수 있다는 의미.

6. story : 백로그에 담겨질 요구사항으로, 고객의 요구사항을 간단한 시나리오로 표현한 것.

7. system architecture : 시스템 아키텍처란 시스템 내부에서 각각의 하위 시스템들이 어떠한 관계로 상호 작용하는지 파악할 수 있도록 구성이나 동작 원리를 표현한 것.

8. core : 코어란 각종 연산 수행하는 CPU의 핵심 요소. 코어 개 갯수에 따라 싱글 코어, 듀얼코어, 트리플 코어 등으로 구분하며 갯수가 많을수록 속도가 빨라짐.

9. Operating system(OS) : 운영체제는 컴퓨터 시스템의 자원들을 효율적으로 관리하며, 사용자가 컴퓨터를 편리하고 효율적으로 사용할 수 있는 환경을 제공하는 소프트웨어.

10. middleware : 미들웨어는 주로 상하 관계나 동종 관계로 구분할 수 있는 프로그램들 사이에서 매개 역할을 하거나 프레임워크 역할을 하는 일련의 중간 계층 프로그램. 일반적으로 응용 프로그램과 운영 체계의 중간 계층에 위치하여 사용자에게 시스템 하부에 존재하는 하드웨어, 운영 체계, 네트워크에 상관없이 서비스를 제공한다.

11. availability : 가용성이란 사용이 요구될 때 한 소프트웨어가 지정된 시스템 기능을 수행 할 수 있는 능력. 총 운영 시간에 대한 시스템 가동 시간의 비율.

12. memory leak : 메모리 누수란 응용 프로그램이 더 이상 사용하지 않는 메모리를 반환하지 않고 계속 점유하고 있는 현상. 성능 저하나 메모리 고갈 등의 장애 발생 가능.

13. pair programming :짝 프로그래밍은 다른 사람과 함께 프로그래밍을 수행함으로 개발에 대한 책임을 공동으로 나눠 갖는 환경 조성.

14. open source : 오픈 소스(공개 소스)란 일정한 이용 조건을 지키면 누구나 개량하고, 재배포할 수 있도록 무상으로 공개되는 소스 코드(source code).

15. TOC(Total cost of ownership) : 총 소유 비용이란 PC나 서버 등의 도입 비용뿐만 아니라 향상(upgrade)이나 유지 보수, 교육 연수 등과 같이 도입 후에 드는 여러 가지 비용을 포함한 컴퓨터 시스템의 총비용.