

# 교육 과정 소개서.

---

한 번에 끝내는 유니티&C# 게임 개발 초격차 패키지



## 강의정보

강의장	온라인 강의   데스크탑, 노트북, 모바일 등
수강 기간	평생 소장
상세페이지	<a href="https://fastcampus.co.kr/dev_online_unitygame">https://fastcampus.co.kr/dev_online_unitygame</a>
강의시간	152시간
문의	<a href="#">고객센터</a>

## 강의특징

나만의 속도로 **낮이나 새벽이나 내가 원하는 시간대에** 나의 스케줄대로 수강

원하는 곳 어디서나 **시간을 쪼개 먼 거리를 오가며 오프라인 강의장을 찾을 필요 없이 어디서든 수강**

무제한 복습 **무엇이든 반복적으로 학습해야 내것이 되기에 이해가 안가는 구간 몇번이고 재생**



## 강의목표

- 유니티로 만들 수 있는 인기 장르 미니게임 6개, 최근 대세인 장르(범서라이크, RPG) 게임 2개까지 총 8개의 게임 개발 실습 프로젝트를 진행합니다.
- 대형 게임사/ 인디 게임 개발자, 게임사 대표, IT기업 개발팀장, 유니티 전문 엔지니어까지 어벤저스 강사진을 모두 모았습니다.
- 개발이 처음인 사람부터 실전 연습이 필요한 사람까지, 실력에 따라 맞춤형 학습이 가능합니다. 탄탄한 기본기에 심화 기능을 더해 재미부터 완성도까지 두 마리 토끼를 잡을 수 있습니다.

## 강의요약

- C# 문법, 유니티 아키텍처 설계, 게임물리&수학, 디자인패턴, UNIRX, 플랫폼 빌드 등 게임을 만들 때 필요한 핵심 개념을 모두 학습 가능합니다.
- 유니티 게임 개발의 기반이 되는 게임&컴퓨터공학 지식, C#, 유니티 엔진 기초부터 게임 만드는 재미를 느낄 수 있는 미니게임, 상용화 레벨 게임 개발까지 실습 할 수 있도록 구성했습니다.
- 많은 플레이어들이 열광하는 게임에는 뭔가 특별한 것이 있다! 중독성 있는 미니게임부터 인기있는 장르의 게임까지 함께 개발해 봅시다.
- 게임 개발 최전선에서 동시대의 게임을 만들며 활발하게 활동하고 있는 진짜 현업 게임 개발자의 생생한 노하우를 배워보세요.



## 강사

이재현	과목	- C#, UNITY 담당
	약력	- 모닛(삼성 C-LAB) 개발팀장 - 아프리카TV 프로그래머 - 위메이드 클라이언트 프로그래머
카골드	과목	- 미니게임 개발 실습 담당
	약력	- 대인배들 프로그래머/PM - 플레이하드 프로그래머 - 엘릭서 프로그래머
송기호	과목	- 뱀서라이크 개발 실습 담당
	약력	- NC소프트 클라이언트 프로그래머 - 네오위즈 클라이언트 프로그래머 - 컴투스 프로그래머
고준일	과목	- 게임공학 강의 담당
	약력	- 게임제작사 모카포트 대표 - Unity Certified Instructor - 호서대 게임학과 겸임교수 - 명지전문대 소프트웨어콘텐츠학과 겸임교수
김원섭	과목	- RPG 개발 실습 담당
	약력	- 조이시티 클라이언트 프로그래머 - 그레비티 네오싸이언 게임 프로그래머 - 아이피플스 게임 프로그래머

## CURRICULUM

## 01.

## C# 기초

파트별 수강시간 04:24:25

<b>CH01. 오리엔테이션</b>
01. 커리큘럼 개요
<b>CH02. 환경 설정</b>
01. 환경설정, Hello World!
<b>CH03. 변수, 메서드, 클래스</b>
01. 객체 지향 프로그래밍 그림으로 이해하기
02. class 활용
03. 메서드 활용
<b>CH04. 배열, List</b>
01. 배열에 데이터 담기
02. 리스트에 데이터 담기
<b>CH05. for</b>
01. for 문법과 활용
<b>CH06. foreach</b>
01. foreach 문법과 활용
<b>CH07. 데이터 형식</b>
01. 데이터형식
02. 열거 형식 (enum)
03. 형변환
<b>CH08. Value Type, Reference Type</b>
01. Value Type, Reference Type

## CURRICULUM

## 01.

## C# 기초

파트별 수강시간 04:24:25

<b>CH09. Call by value, Call by reference</b>
01. Call by value, Call by reference
<b>CH10. 데이터 표현</b>
01. 문자열 가공 (Contains, Replace, Trim, ToLower, 기타)
02. 문자열 가공 (string.format, ToString(), \$, +, StringBuilder)
03. 날짜와 시간 가공(DateTime)
<b>CH11. 연산자</b>
01. 산술연산자의 형변환 (+, -, *, /, %), 증감 연산자(x++, ++x)
02. 관계 연산자(==), 논리연산자(  , &&), 부정(!), 할당 연산자(+=)
03. binary, BitArray
<b>CH12. 흐름 제어</b>
01. if, switch, 삼항연산자(조건연산자)
02. while, do while, break, continue, 2중 for문
<b>CH13. null 처리</b>
01. null 검사하기, 병합 컨트롤(?, ?? 의 활용), nullable
<b>CH14. 예외처리</b>
01. try catch
<b>CH15. 프로퍼티</b>
01. 프로퍼티 (Property)
<b>CH16. Visual studio</b>
01. Visual studio 파일 찾기 및 단축키




---

CURRICULUM

02.

C# 심화

파트별 수강시간 04:54:07

---

<b>CH01. 오리엔테이션</b>
01. 오리엔테이션
<b>CH02. 디버깅</b>
01. 디버깅
<b>CH03. 메서드</b>
01. 오버로딩(overloading)
02. 가변 파라미터 (params), 선택적 인수, 명명된 인수
03. 대리자 (Delegate, Action)
<b>CH04. 클래스</b>
01. 생성자(constructor), readonly, 한정자(private, public), this.
02. 상속 (Overriding)
03. 상속 (Up, Down Casting)
04. DI (Dependency Injection)
05. static
06. deepcopy
07. 서브클래스
08. 인터페이스, 추상클래스



CURRICULUM

02.

C# 심화

파트별 수강시간 04:54:07

<b>CH05. 람다식</b>
01. 람다식 (Lambda)
<b>CH06. 링큐</b>
01. 링큐 (Linq)
<b>CH07. 제네릭</b>
01. 제네릭 (Generic)
<b>CH08. 비동기</b>
01. 비동기 프로그래밍 async
<b>CH09. 리플렉션</b>
01. 리플렉션 (Reflection)
<b>CH10. DTO</b>
01. DTO (MVC, Entity, Repository, Nuget)
<b>CH11. operator</b>
01. operator
<b>CH12. Enumerator</b>
01. Enumerator
<b>CH13. 기타</b>
01. 기타(object, dictionary, enum컨트롤, struct, consolas)
02. 기타(상속(new, sealed, override), goto, 경렬, 다차원배열, 가변배열)
03. 기타(class (종료자(Finalizer), this() 생성자, 분할 클래스), 프로퍼티 (init, required), 튜플)
04. 기타(using dispose, 인덱서, try catch 추가 설명, 확장메서드)



CURRICULUM

03.

유니티 엔진 기초

파트별 수강시간 09:12:11

<b>CH01. 오리엔테이션</b>
01. 커리큘럼 개요
<b>CH02. 환경 설정</b>
01. 유니티 환경설정
<b>CH03. 유니티 툴 알아보기</b>
01. 유니티 툴 알아보기
<b>CH04. Hello World!</b>
01. 오브젝트 생성과 Hello World!
<b>CH05. 카메라</b>
01. 카메라
<b>CH06. 프리팹 만들기과 로드</b>
01. 프리팹 만들기과 로드
<b>CH07. 오브젝트 컨트롤</b>
01. 충돌 판정하기
02. 입력 처리 및 좌표 변환
03. 오브젝트 간격체크
04. Raycast, LayerMask
05. 삼각함수로 오브젝트 이동
06. 선형보간으로 오브젝트 이동
07. 코루틴
08. 로컬, 글로벌 좌표계
09. FPS 움직임 만들기
10. NavMesh
11. 카메라 셰이크
<b>CH08. 캐릭터 생성과 관리</b>
01. 애니메이션
02. 사운드
03. 파티클시스템, 머터리얼
04. FSM
05. 오브젝트 풀
<b>CH09. UGUI</b>
01. 이미지 최적화
03. UGUI 컴포넌트
04. UGUI 해상도 대응
05. Grid Layout, 스크롤
06. 인벤토리
07. 메시지 팝업 만들기
08. 팝업 전환

## CURRICULUM

## 04.

## 유니티 엔진 심화

파트별 수강시간 08:14:56

CH01. 오리엔테이션
01. 커리큘럼 개요
CH02. 프로젝트 아키텍처
01. Awake, Start
02. Init 초기화 씬 만들기
03. Init 연속적인 호출
04. Singleton
05. 씬 로드 및 관리
06. 씬 바로 실행하기
07. 네트워크 연동 환경 구축
08. Dev, Stage, Release 1
09. Dev, Stage, Release 2
10. 초기 Config 패킷 - 코루틴
11. 초기 Config 패킷 - 콜백
12. 초기 Config 패킷 - 테스트 권한
13. 점검
14. 업데이트
15. 로컬 저장, 데이터 자료구조 1
16. 로컬 저장, 데이터 자료구조 2
17. SNS 연동, 게스트 모드
18. 보안, 암호화
19. Localization 매니저
20. Language 매니저
CH03. 로깅
01. 로깅
CH04. 기타기능
01. 인디게이터, 프레임표시, 타임스케일, Resources
CH05. 에셋번들
01. 에셋번들
CH06. 신경 덜 쓰는 코드
01. 신경 덜 쓰는 코드
CH07. 개발 철학
01. 개발 철학
CH08. 강의 종료
01. 종강

## CURRICULUM

## 05-1.

# 미니게임 프로젝트 (클릭클릭대소동)

파트별 수강시간 02:52:46

<b>CH01. 소개</b>
01. 개요
02. 작업 계획
<b>CH02. 기능 작업</b>
01. 기반 구축
02. 입력 시스템
03. 노트그룹 리소스
04. 노트 생성 및 배치
05. 노트 그룹 애니메이션
06. 노트 그룹 조작
07. 랜덤 사과 생성
08. 스코어 기록
09. UI
10. 가변화 1
11. 가변화 2
12. 노트 그룹 동적 생성
13. 게임 종료
14. 폴리싱
<b>CH03. 마무리</b>
01. 마무리
<b>CH04. 폴리싱</b>
01. 아이템 이동 시 트레일 렌더러
02. 아이템 획득 시 파티클 이펙트
03. UI 연출
<b>CH05. 마무리</b>
01. 업데이트 방향성

## CURRICULUM

## 05-2.

# 미니게임 프로젝트 (점프래빗)

파트별 수강시간 05:22:18

<b>CH01. 개요</b>
01. 개요
02. 프로젝트 세팅
<b>CH02. 개발</b>
01. 기반 구축
02. 리지디바디와 점프
03. 애니메이션 기반
04. 애니메이션 구현
05. 프리팹 배리언트
06. 플랫폼
07. 플랫폼 배치 알고리즘
08. 플랫폼 간격 알고리즘
09. 카메라_1
10. 스크립터블 오브젝트와 데이터베이스
11. 점수 기반
12. 점수 연출
13. 보너스 점수 1
14. 보너스 점수 2
15. 아이템
16. 사운드
17. 폴리싱 1
18. 폴리싱 2
19. 폴리싱 3
20. 폴리싱 4
<b>CH03. 마무리</b>
01. 빌드

## CURRICULUM

## 05-3.

## 미니게임 프로젝트 (플래피 슈퍼스타C)

파트별 수강시간 04:07:16

CH01. 소개
01. 개요
02. 개발 준비
CH02. 기능 작업
01. 기반 구축
02. 플레이어
03. 장애물
04. 장애물 - 생성
05. 장애물 - 위치 조절
06. 장애물 - 통과
07. 레벨 디자인
08. 게임 오버
09. 카메라
10. 배경 꾸미기
11. 배경 꾸미기 - 소녀팬
12. 배경 꾸미기 - 슬로건
13. UI
14. 사운드
15. 폴리싱 - 사운드
16. 폴리싱 - 시스템
17. 폴리싱 - 연출
CH03. 마무리
01. 해상도 대응 및 빌드



CURRICULUM

05-4.

**미니게임  
프로젝트(밤산책)**

파트별 수강시간 03:36:05

CH01. 소개
01. 기획 브리핑
CH02. 기능 작업
01. 기반 작업
02. 플레이어 1 - 기반 구축
03. 플레이어 2 - 애니메이션
04. 맵 - 꾸미기
05. 맵 - 허용 영역
06. 몬스터 1 - 구축
07. 몬스터 2 - 기능 구현
08. Light 2D 1 - 구축
09. Light 2D 2 - 적용
10. 플레이어 3 - 손전등과 발자국
11. 아이템 - 강통칸
12. 아이템 - 열쇠
13. 사운드
14. 폴리싱
15. 승리 및 재시작

## CURRICULUM

05-5.

# 미니게임 프로젝트(리듬 오브 셰이크댄서)

파트별 수강시간 03:50:21

CH01. 소개
01. 개요
CH02. 기능 작업
01. 기반 작업
02. 비트 매니저
03. 비트 UI
04. 캐릭터 1 - 이동
05. 캐릭터 2 - 애니
06. 블록 1 - 기반 구축
07. 블록 2 - 기능 구현
08. 블록 3 - 레벨 디자인
09. 배경 꾸미기
10. 점수 시스템 1 - 기반 구축
11. 점수 시스템 2 - 연출
12. 비트 정확도 구현
13. UI
14. 엔딩
15. 사운드
16. 폴리싱

## CURRICULUM

## 05-4.

## 미니게임 프로젝트(투 핑거 데스펀치)

파트별 수강시간 04:24:36

CH01. 소개
01. 개요
CH02. 기능 작업
01. 기반 구축
02. 타일 시스템 - 기반
03. 타일 시스템 - 캐릭터
04. 타일 시스템 - 전투
05. 인게임 플로우 - UI
06. 인게임 플로우 - 구현
07. 전투 플로우
08. 보상
09. 업그레이드
10. 업그레이드 - UI
11. 결과창 및 스텟 UI
12. 마무리

## CURRICULUM

## 06.

## 뱀서라이크 (기본구현)

파트별 수강시간 14:08:25

<b>CH01. 소개</b>
01. 강사소개와 개발할 게임 소개
<b>CH02. 기본 구현</b>
01. 만드는 게임에 구현 설계
02. 게임 매니저 설계
03. 게임 매니저 구현
04. 게임 스테이지 구현
05. 게임의 시작과 끝
06. 내 캐릭터의 설계
07. 카메라
<b>CH03. 캐릭터 이동</b>
01. 내 캐릭터의 이동
<b>CH04. 기본 관리</b>
01. 게임의 오브젝트 관리
<b>CH05. 캐릭터 공격</b>
01. 내 캐릭터의 스킬 관리
02. 스킬의 개별 구현
<b>CH06. 캐릭터 스킬</b>
01. 스킬의 데이터 설계
02. 스킬의 데이터 드리븐 구현
03. 내 캐릭터의 스킬 확장

## CURRICULUM

## 06.

# 뱀서라이크 (기본구현)

파트별 수강시간 14:08:25

<b>CH07. 몬스터</b>
01. 몬스터 관리와 관련된 설계
02. 몬스터 데이터 구성
03. 몬스터 소환
<b>CH08. 상호작용</b>
01. 내 캐릭터와 적의 상호작용 설계
02. 상호작용 구현
<b>CH09. 캐릭터의 성장</b>
01. 아이템
02. 아이템 드랍
03. 아이템 줍기
04. 레벨 증가와 기술 증가
<b>CH10. 게임 UI</b>
01. UI 화면에 대한 설계
02. HUD 화면에 대한 구현
03. 게임에 필요한 UI 구현
04. 게임 내 오브젝트의 UI 구현
<b>CH11. 이펙트, 사운드</b>
01. 사용하는 스킬에 이펙트 적용
02. 게임의 BGM과 사운드 적용

## CURRICULUM

## 07.

## 뱀서라이크(확장 구현)

파트별 수강시간 13:17:03

<b>CH01. 게임의 확장 소개</b>
01. 게임의 확장 소개
<b>CH02. 게임 구조 변경</b>
01. 게임의 상태 변경 설계
02. 게임의 상태 변경 구현
<b>CH03. 스킬</b>
01. 레벨업시 게임 스킬 선택
02. 사용 스킬에 대한 설계
03. 사용 스킬 시스템에 대한 구현
<b>CH04. 홈</b>
01. 홈에 필요한 기능 설계
02. 홈 UI 기본 구현
03. 인벤토리 기능 구현
04. 장착 기능 구현
05. 장착된 아이템의 능력치
06. 상점 UI
07. 상점을 위한 재화 만들기
08. 상점 기능 구현
<b>CH05. 강화</b>
01. 강화 설계
02. 아이템 강화 UI
03. 아이템 강화 구현
<b>CH06. 인벤토리</b>
01. 획득 아이템의 저장 구현
<b>CH07. 보스</b>
01. 보스에 대한 설계
02. 보스 등장으로 인한 시스템 변경
03. 보스 에셋 준비
04. 보스의 애니메이션 구현
05. 보스의 스킬 구현
<b>CH08. 게임 회사 이야기</b>
01. 네트워크 게임을 만들 때 주의점
02. 사업화 게임을 만든 이후 판매까지
03. 프로젝트 출시까지의 성공과 실패 경험
02. 아이템 강화 UI

## CURRICULUM

## 08.

## 게임 공학

파트별 수강시간 36:29:39

## CH01. 오리엔테이션

01. 과정 소개

## CH02. 게임 수학

01. 유니티 수학이란?

02. 삼각함수 이론 1

03. 삼각함수 이론 2

04. 삼각함수 실습 1

05. 삼각함수 실습 2

06. 삼각함수 실습 3

07. 벡터 이론

08. 벡터 실습 1

09. 벡터 실습 2

10. 벡터 실습 3

11. 벡터 실습 4

12. 좌표계 이론

13. 좌표계 실습 1

14. 좌표계 실습 2

15. 행렬 이론

16. 행렬 실습

17. 사원수 이론 1

18. 사원수 이론 2

19. 사원수 실습

## CH03. 게임 물리

01. 유니티 물리란

02. 리지드 바디와 콜라이더 1

03. 리지드 바디와 콜라이더 2

04. 리지드 바디와 콜라이더 3

05. 리지드 바디와 콜라이더 4

06. 리지드 바디와 콜라이더 5

07. 조인트와 관절 1

08. 조인트와 관절 2

09. 조인트와 관절 3

10. 조인트와 관절 4

11. 조인트와 관절 5

12. 캐릭터 컨트롤러 1

13. 캐릭터 컨트롤러 2

14. 휠 콜라이더 1

15. 휠 콜라이더 2

## CURRICULUM

## 08.

## 게임 공학

파트별 수강시간 36:29:39

## CH04. 자료구조와 알고리즘

01. 자료구조와 알고리즘이란

02. 예제 준비

03. Array List

04. Hashtable

05. Array List와 Hashtable

06. Linked List-1

07. Linked List-2

08. Linked List-3

09. Linked List-4

10. Linked List-5

11. Linked List-6

12. Stack-1

13. Stack-2

14. Stack-3

15. Stack-4

16. Queue

17. Recursive Function

18. Binary Search

19. Binary Search Tree

20. AVL Tree - 1

21. AVL Tree - 2

22. AVL Tree - 3

23. Heap Tree - 1

24. Heap Tree - 2

25. Graph - 1

26. Graph - 2

27. Graph - 3

28. Graph - 4

29. Dijkstra - 1

30. Dijkstra - 2

31. Dijkstra - 3

32. AStar - 1

33. AStar - 2

34. AStar - 3

35. AStar - 4

36. AStar - 5



CURRICULUM

08.

게임 공학

파트별 수강시간 00:00:00

<b>CH05. 디자인 패턴</b>
01. 디자인 패턴이란
02. 싱글톤 패턴 이론
03. 싱글톤 패턴 실습
04. 오브젝트 풀 패턴 이론
05. 오브젝트 풀 패턴 실습 1
06. 오브젝트 풀 패턴 실습 2
07. 상태 패턴 이론
08. 상태 패턴 실습
09. 옵저버 패턴 이론
10. 옵저버 패턴 실습
11. UniRX 이론
12. UniRX 실습 1
13. UniRX 실습 2
14. UniRX 실습 3
<b>CH06. 플랫폼 빌드 및 배포</b>
01. Windows 빌드 이론
02. Windows 빌드 실습 1
03. Windows 빌드 실습 2
04. Android 앱 빌드 이론
05. Android 앱 빌드 실습
06. iOS 앱 빌드 실습
07. WebGL 빌드 실습
<b>CH07. 형상관리</b>
01. 형상관리란 1
02. 형상관리란 2
03. JIRA를 이용한 프로젝트 관리 1
04. JIRA를 이용한 프로젝트 관리 2
05. Notion을 이용한 프로젝트 관리
06. Git을 이용한 소스 관리 이론
07. Git을 이용한 소스 관리 실습 1
08. Git을 이용한 소스 관리 실습 2
09. Git을 이용한 소스 관리 실습 3
10. Git을 이용한 소스 관리 실습 4
11. PlasticSCM을 이용한 소스 관리 이론
12. PlasticSCM을 이용한 소스 관리 실습 1
13. PlasticSCM을 이용한 소스 관리 실습 2



CURRICULUM

**09.**  
**RPG**

파트별 수강시간 37:24:21

<b>CH01. 소개</b>
01. 강사소개와 개발할 게임 소개
<b>CH02. 프로젝트 및 구현 목록</b>
01. 프로젝트 및 구현 목록 설명
<b>CH03. 기본 설계</b>
01. 프로젝트 뼈대 설계 - 1
02. 프로젝트 뼈대 설계 - 2
03. 프로젝트 뼈대 설계 - 3
<b>CH04. 작업 편의를 위한 간단한 기능구현tip</b>
01. 작업 편의를 위한 간단한 기능 구현tip
<b>CH05. 스테이지 기본 구현</b>
01. 스테이지 기본 구현 - 1
02. 스테이지 기본 구현 - 2
03. 스테이지 기본 구현 - 3
04. 스테이지 기본 구현 - 4
05. 스테이지 기본 구현 - 5
<b>CH06. 카메라 기본 구현</b>
01. 카메라 구현
<b>CH07. 캐릭터 구현</b>
01. 캐릭터 구현 - 1
02. 캐릭터 구현 - 2
03. 캐릭터 구현 - 3
04. 캐릭터 구현 - 4
<b>CH08. 길찾기 구현</b>
01. 길찾기 구현 - 1
02. 길찾기 구현 - 2
03. 길찾기 구현 - 3
04. 길찾기 구현 - 4



CURRICULUM

**09.**  
**RPG**

파트별 수강시간 37:24:21

<b>CH09. 오브젝트 구현</b>
01. 오브젝트 구현 - 1
02. 오브젝트 구현 - 2
03. 오브젝트 구현 - 3
04. 오브젝트 구현 - 4
<b>CH10. 구현 플레이</b>
01. 구현 플레이 - 1
02. 구현 플레이 - 2
<b>CH11. 스테이지 심화 구현</b>
01. 스테이지 심화 구현 - 1
02. 스테이지 심화 구현 - 2
03. 스테이지 심화 구현 - 3
04. 스테이지 심화 구현 - 4
<b>CH12. 구현 플레이</b>
01. 구현 플레이
<b>CH13. 스킬 구현</b>
01. 스킬 구현 - 1
02. 스킬 구현 - 2
03. 스킬 구현 - 3
04. 스킬 구현 - 4
<b>CH14. 캐릭터 구현</b>
01. 플레이어 캐릭터 - 1
02. 플레이어 캐릭터 - 2
03. 적군 캐릭터 - 1
04. 적군 캐릭터 - 2
05. 적군 캐릭터 - 3
06. 적군 캐릭터 - 4



CURRICULUM

**09.**  
**RPG**

파트별 수강시간 37:24:21

<b>CH15. 전투 시스템 연계</b>
01. 전투 시스템 연계 - 1
02. 전투 시스템 연계 - 2
03. 전투 시스템 연계 - 3
04. 전투 시스템 연계 - 4
05. 전투 시스템 연계 - 5
<b>CH16. 구현 플레이</b>
01. 구현 플레이
<b>CH17. UI 구현</b>
01. UI 구현 설명
02. UI 구현 - 1
03. UI 구현 - 2
04. UI 구현 - 3
05. UI 구현 - 4
06. UI 구현 - 5
07. UI 구현 - 6
<b>CH18. 웨이더 기본 설명</b>
01. 웨이더 기본 설명 - 1
02. 웨이더 기본 설명 - 2
03. 웨이더 구현 - 1
04. 웨이더 구현 - 2
05. 웨이더 구현 - 3
<b>CH19. 마무리 작업</b>
01. 마무리 작업 - 1
02. 마무리 작업 - 2
03. 마무리 작업 - 3
04. 마무리 작업 - 4
<b>CH20. 구현 플레이</b>
01. 구현 플레이

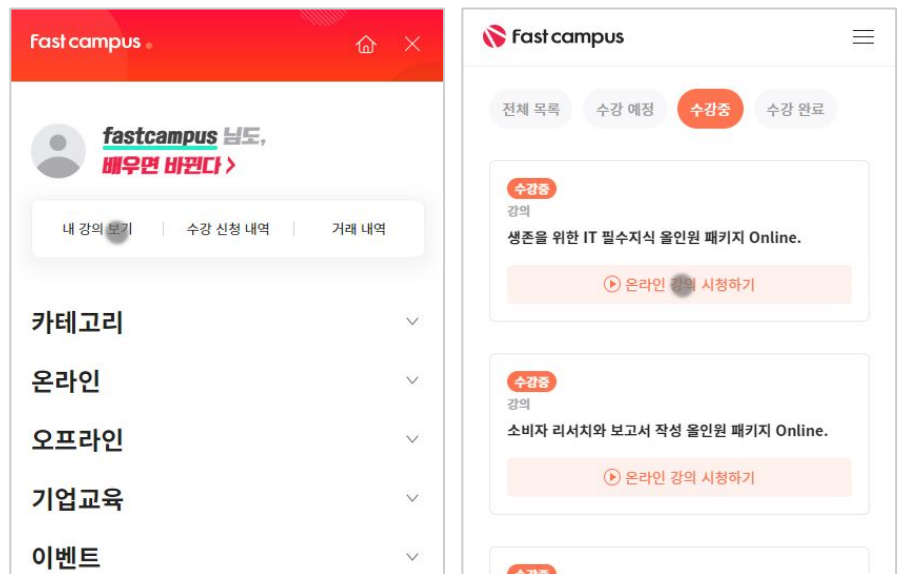


## 주의 사항

- 상황에 따라 사전 공지 없이 할인이 조기 마감되거나 연장될 수 있습니다.
- 패스트캠퍼스의 모든 온라인 강의는 아이디 공유를 금지하고 있으며 1개의 아이디로 여러 명이 수강하실 수 없습니다.
- 별도의 주의사항은 각 강의 상세페이지에서 확인하실 수 있습니다.

## 수강 방법

- 패스트캠퍼스는 크롬 브라우저에 최적화 되어있습니다.
- 사전 예약 판매 중인 강의의 경우 1차 공개일정에 맞춰 '온라인 강의 시청하기'가 활성화됩니다.



## 환불 규정

- 온라인 강의는 각 과정 별 '정상 수강기간(유료수강기간)'과 정상 수강기간 이후의 '복습 수강기간(무료수강기간)'으로 구성됩니다.
- 환불금액은 실제 결제금액을 기준으로 계산됩니다.

수강 시작 후 7일 이내	100% 환불 가능 (단, 수강하셨다면 수강 분량만큼 차감)
수강 시작 후 7일 경과	정상(유료) 수강기간 대비 잔여일에 대해 환불규정에 따라 환불 가능

※ 강의별 환불규정이 상이할 수 있으므로 각 강의 상세페이지를 확인해 주세요.