1. **스위프트 특성**
2. 객체지향 (OOP)

클래스와 객체, 클래스를 정의하고 생성하여 할당할때 우리는 그것을 인스턴스라 부른다.  let instance = Class()

1. 함수형 프로그래밍

|  |  |  |
| --- | --- | --- |
|  | **명령형 프로그래밍** | **함수형 프로그래밍** |
| 프로그래머가 초첨을 두는 곳 | 작업수행 알고리즘 상태의 변경 추적 | 원하는 정보 필요한 변환 |
| 상태변경 | 중요 | 없음 |
| 실행순서 | 중요 | 낮은 중요도 |
| 주요 흐름 제어 | 제어 구문(반복문, 조건문)  함수(메서드) 실행 | 순환(재귀) 함수 호출등의 함수 호출로 제어 |
| 주요 조작 단위 | 클래스나 구조체의 인스턴스 | 함수 |

* 여러 가지 연산 처리 작업이 동시에 일어나는 프로그램을 만들기 쉬움
* 멀티 코어 혹은 여러 개 연산 프로세서를 사용하는 시스템에서 효율적임
* 상태변화에서 자유로워지므로 순수하게 기능 구현에 집중할수 있음

1. 프로토콜 지향

참조타입의 참조 문제에서 조금 더 자유로울수 있고 다중상속이 불가능한 한계를 뛰어넘을 수 있으며, 더 나은 추상화 매커니즘을 구현할 수 있음.