
스프레드시트를 활용한 데이터 체계화

스프레드시트는 누구나 읽을 수 있게

1. 통합 문서 작성은 누구나 쓸 만하게 만든다
2. 작업 전 미리 스프레드시트로 수행할 작업을 계획하고 사용법을 정돈해 둔다
3. 다른 사람이 스프레드 시트를 어떻게 볼지 데이터를 어떻게 볼지 다른 사람의 관점을 고려해서 작성한다

되도록 숫자는 피할 것

1. 스프레드시트에서 숫자를 직접 입력하기보다는 참조를 활용하자

- 간단한 한자리 숫자라 해도 리스트 안의 모든 계산에 특정한 숫자를 직접 하는 것을 지양한다
- 참조 시트를 따로 만들어 셀에 레이블을 지정하고 변수를 넣는다
- 해당 변수가 필요한 모든 곳에는 변수명으로 참조를 지정한다
- 변수와 연결된 숫자가 변경 될 때마다 이를 사용하는 모든 방정식이 자동으로 업데이트 된다

레이블을 붙이자

1. 레이블이 수식의 맥락을 알려준다

- 수식이 무엇을 하려는 것인지 알 수 있다
- 오류가 발생시 이유를 파악하는데 큰 도움이 된다
- 메모나 댓글을 달아도 도움이 된다
- 셀 하나에 너무 많은 수식을 만들지 않는다 수식을 여러 부분으로 나나고 레이블을 지정하면 문제 발생시 훨씬 쉽게 해결할 수 있다

속성은 열에 객체는 행에

1. 게임에는 데이터 객체에 내재된 속성보다도 데이터 객체 자체가 더 많다
2. 데이터 객체는 각기 한 행 씩 배정해서 세로 방향으로 여러 페이지에 걸쳐 정리하되, 속성은 한 페이지 너비로 제한되도록 체계화 한다

색상 코딩으로 시각적 체계화

1. 색상 코딩 체계 예제

- 헤더는 검정 바탕에 흰색 글씨로 : 일반적으로 1행에 놓이는 헤더는 속성이나 범주의 메인 레이블로 쓰인다
- 입력 데이터는 황갈색 : 데이터 입력 셀은 튜닝하면서 계속 바뀔 부분이다
- 수식은 녹색 : 통합 문서의 다른 위치에 있는 데이터를 참조하거나 NOW() 같은 독립 실행형 계산을 수행하는 수식이나 함수가 들어간다
- 하위 레이블은 파랑 : 하위 레이블은 하위 범주에 사용된다. 대체로 데이터 객체명을 파란색으로 표시해 데이터의 나머지 부분과 구분한다
- 고정 텍스트는 흰색 바탕에 검정색 글씨 : 고정 데이터는 다른 곳에서 참조용으로 가져왔을 경우가 많아 튜닝 등의 목적으로 변경해서는 안된다

불필요한 열, 행, 빈 셀 금지

1. 테이블의 정보를 시각적으로 보기 좋게 하기 위해 빈 열이나 행 또는 셀을 추가하지 않는다
 - 빈 행이 있을 경우 계산 흐름이 중단된다
 - 스프레드시트의 기능이 제대로 작동되지 않는다

데이터 객체 종류별로 시트를 분리하자

1. 참조 시트

- 서로 다른 시트에서 여러 번 사용하게 되는 입력값들의 은행
- 시트를 복잡하게 만드는 변수는 참조 시트에 저장해서 시트를 간단하게 만든다

2. 개요 시트

- 통합 문서 맨 처음에 있는 시트로서 가이드, 시작, 도움말 같은 레이블이 붙는다
- 각 시트의 용도를 설명하고 그 밖에 필요한 온갖 지침을 보여줄 수 있다

3. 출력/시각화 시트

- 많은 데이터 객체를 함수와 수식으로 데이터를 분석하거나 차트로 시각화 할 수 있다

목차

1.

-

-

2.

-

-

3.

-

-