

## FLS for Developers

- Apex classes do not enforce FLS
  - Runs in system context
- Visualforce pages enforce FLS
  - Runs in user context
  - Does not enforce FLS for dereferenced fields
    - `{!Account.AnnualRevenue}` = yes because it is an sObject reference
    - `{!wRevenue.field}` = no because it is a wrapper object reference

- Wrapper:

- `Random_Sensitive_Object_1__c r; // Salesforce sObject`
- `wRandom_Sensitive_Wrapper wR; // Custom wrapper object`
- `wR.field = r.Sensitive_Field__c;`

## When does the Platform stop respecting FLS?

When sObject's field is assigned to a primitive

Apex:

```
Random_Sensitive_Object_1__c r; // Salesforce sObject
wRandom_Sensitive_Wrapper wR; // Custom wrapper object
wR.field = r.Sensitive_Field__c; //field is a generic string type
```

Visualforce:

- `<apex:OutputText value="{!r.Sensitive_Number__c}" /> <-- FLS RESPECTED`
- `<apex:OutputText value="{!wR.Sensitive_Number}" /> <-- FLS IGNORED`