`rustc_attrs`

This feature has no tracking issue, and is therefore internal to the compiler, not being intended for general use.

Note: `rustc_attrs` enables many rustc-internal attributes and this page only discuss a few of them.

The `rustc_attrs` feature allows debugging rustc type layouts by using `#[rustc_layout(...)]` to debug layout at compile time (it even works with `cargo check`) as an alternative to `rustc -Z print-type-sizes` that is way more verbose.

Options provided by `#[rustc_layout(...)]` are `debug`, `size`, `align`, `abi`. Note that it only works on sized types without generics.

Examples

```
#![feature(rustc_attrs)]

#[rustc_layout(abi, size)]
pub enum X {
    Y(u8, u8, u8),
    Z(isize),
}
```

When that is compiled, the compiler will error with something like

```
error: abi: Aggregate { sized: true }
 --> src/lib.rs:4:1
  |
4 | / pub enum T {
5 | |     Y(u8, u8, u8),
6 | |     Z(isize),
7 | | }
  | |_^

error: size: Size { raw: 16 }
 --> src/lib.rs:4:1
  |
4 | / pub enum T {
5 | |     Y(u8, u8, u8),
6 | |     Z(isize),
7 | | }
  | |_^

error: aborting due to 2 previous errors
```