# سخن اول

ما به دو دلیل Clean Code می خونیم دلیل اول اینکه برنامه نویس هستیم دلیل دوم اینکه قصد داریم برنامه نویس بهتری بشیم.

# [بخش اول](https://virgool.io/@NimaShirinzadeh/clean-code-%D8%A8%D8%AE%D8%B4-%D8%A7%D9%88%D9%84-dgzw9ik0zusd)

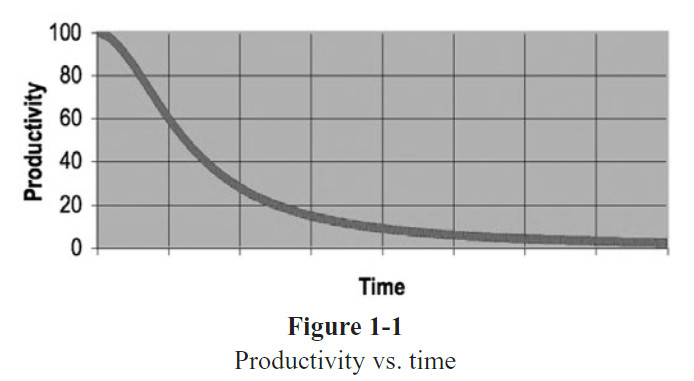
# چرا Clean Code

ما مجبوریم کد تمیز بنویسیم چون با کد کثیف رفع باگ، ارائه فیچر یا ارائه نسخه جدید سخت تر و سختر میشه و با بزرگ شدن پروژه هی کار سخت تر میشه و فاصله بین ریلیزها بیشتر میشه تا در نهایت به نقطه ای میرسیم که مجبوریم مرگ پروژه رو اعلام کنیم.

# چرا کد کثیف می نویسیم

چون ددلاین از آنچه فکر می کنیم به ما نزدیکتر است. مدیران رو باید به اهمیت کد تمیز آگاه کنیم و در برآوردهامون زمان تمیز کردن یا تمیز نوشتن رو هم در نظر بگیریم.

# نمودار نسبت بازده به زمان بر اساس یک کد بد



# [بخش دوم](https://virgool.io/@NimaShirinzadeh/clean-code-%D8%A8%D8%AE%D8%B4-%D8%AF%D9%88%D9%85-i3dth5q3dinc)

هدف این بخش اینه که کدهایی که می نویسیم Quick skim‌ باشه و خوندن کدها مثل خوندن روزنامه راحت و روان باشه.

خوب اول اسم تمام متغیرها، فانکشن‌ها و ... باید Meaningful Names باشه.

# تفاوت بین نام معنادار و بدون معنا

نام یک متغیر، کلاس یا فانکشن باید بتونه دلیل وجودی، کاری که داره انجام میده، جا و نحوه ی استفاده شو توصیف کنه.

# نمونه نام های بودن معنا

مخاطب رو دچار سوءتفاهم می کنه

اسامی که در کلماتشون تفاوت کمی وجود داره اسامی مناسبی نیستند.

پرهیز از ایجاد تمایز با انتخاب کلمات متفاوت بدون معنا مثل استفاده از اعداد یا استفاده از کلماتی مثل a، an و the و ...

استفاده از اسامی غیر قابل تلفظ و ناخوانا

برای یک مفهوم چند کلمه انتخاب نشه و یک کلمه چند مفهوم را پشتیبانی نکنه مثلا واژه get همیشه یک مفهوم رو ساپورت کنه

# دو استراتژی برای انتخاب نام

بر اساس راه حل (بهترین استراتژی)

براساس مشکل (تنها مشکل این استراتژی زمانی هست که شخص دیگه ای بخواد روی کد کار کنه بنابراین باید در مورد مشکل از شخصی که توی اون پروژه تجربه داشته سوال بپرسه.)

نکته: زمانی که تمرکز یه کد بیشتر روی حل یک مشکل هست بهتره که اسم از دامنه مشکلات انتخاب بشه.