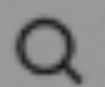




Home



Explore



Notifications



Messages



Grok



Bookmarks



Communities



Premium



Verified Orgs



Profile



More

Post



Kelly Bodwin
@KellyBodwin

Bookmarks

@KellyBodwin

Search Bookmarks



Nicolas Carlo @nicoespeon · Apr 20, 2023

It's common for legacy code to have deeply nested statements

They make code harder to read/think about (cognitive overload) 🤔

Did you know "Guard Clauses" can help you simplify these? Now you know 🙌

Here's a JavaScript example 📄

```
function payAmount(employee, workHours) {  
  let result;  
  if (employee.isSeparated) {  
    result = { amount: 0, reasonCode: "SEP" };  
  } else {  
    if (employee.isRetired) {  
      result = { amount: 0, reasonCode: "RET" };  
    } else {  
      // logic to compute amount  
      payroll.preparePayment(employee.id);  
      employee.startDate = payroll.startDate;  
      const bonuses = employee.bonus(workHours) + payroll.benefit(employee);  
      log.addBonuses(bonuses, employee);  
      result = computeRegularPayAmount(employee, bonuses);  
    }  
  }  
  return result;  
}
```

0:48

26

99

634

72K



Nick Strayer @NicholasStrayer · Jul 28, 2022

Super excited to share with everyone the two ways I've been developing to make building your shiny app's UI faster, easier, and more pleasant: {gridlayout} and {shinyuieditor} (1/5) #RStudioConf #RStats #rstudioconf2022

gridlayout

shinyuieditor



Post



Nicolas Carlo
@nicoespeon

Follow

It's common for legacy code to have deeply nested statements

They make code harder to read/think about (cognitive overload) 🤔

Did you know "Guard Clauses" can help you simplify these? Now you know 🙌

Here's a JavaScript example 📄

```
function payAmount(employee, workHours) {  
  let result;  
  if (employee.isSeparated) {  
    result = { amount: 0, reasonCode: "SEP" };  
  } else {  
    if (employee.isRetired) {  
      result = { amount: 0, reasonCode: "RET" };  
    } else {  
      // logic to compute amount  
      payroll.preparePayment(employee.id);  
      employee.startDate = payroll.startDate;  
      const bonuses = employee.bonus(workHours) + payroll.benefit(employee);  
      log.addBonuses(bonuses, employee);  
      result = computeRegularPayAmount(employee, bonuses);  
    }  
  }  
  return result;  
}
```

0:48

4:15 AM · Apr 20, 2023 · 72.4K Views

26

99

634

123



Post your reply

Reply



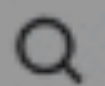
Nicolas Carlo @nicoespeon · Jun 1, 2023

Previous Refa

Messages



Home



Explore



Notifications



Messages



Grok



Bookmarks



Communities



Premium



Verified Orgs



Profile



More

Post



Kelly Bodwin
@KellyBodwin

Bookmarks

@KellyBodwin

Search Bookmarks



Nicolas Carlo @nicoespeon · Apr 20, 2023

It's common for legacy code to have deeply nested statements

They make code harder to read/think about (cognitive overload) 🤔

Did you know "Guard Clauses" can help you simplify these? Now you know 🙌

Here's a JavaScript example 📝

```
function payAmount(employee, workHours) {
  let result;
  if (employee.isSeparated) {
    result = { amount: 0, reasonCode: "SEP" };
  } else {
    if (employee.isRetired) {
      result = { amount: 0, reasonCode: "RET" };
    } else {
      // logic to compute bonuses
      payroll.preparePayment(employee.id);
      employee.startDate = payroll.startDate;
      const bonuses = employee.bonus(workHours) + payroll.benefit(employee);
      log.addBonuses(bonuses, employee);
      result = computeRegularPayAmount(employee, bonuses);
    }
  }
  return result;
}
```

0:48

26

99

634

72K



Nick Strayer @NicholasStrayer · Jul 28, 2022

Super excited to share with everyone the two ways I've been developing to make building your shiny app's UI faster, easier, and more pleasant: {gridlayout} and {shinyuieditor} (1/5) #RStudioConf #RStats #rstudioconf2022

gridlayout

shinyuieditor



Post



Nicolas Carlo
@nicoespeon

Follow

It's common for legacy code to have deeply nested statements

They make code harder to read/think about (cognitive overload) 🤔

Did you know "Guard Clauses" can help you simplify these? Now you know 🙌

Here's a JavaScript example 📝

```
function payAmount(employee, workHours) {
  let result;
  if (employee.isSeparated) {
    result = { amount: 0, reasonCode: "SEP" };
  } else {
    if (employee.isRetired) {
      result = { amount: 0, reasonCode: "RET" };
    } else {
      // logic to compute bonuses
      payroll.preparePayment(employee.id);
      employee.startDate = payroll.startDate;
      const bonuses = employee.bonus(workHours) + payroll.benefit(employee);
      log.addBonuses(bonuses, employee);
      result = computeRegularPayAmount(employee, bonuses);
    }
  }
  return result;
}
```

0:48

4:15 AM · Apr 20, 2023 · 72.4K Views

26

99

634

123



Post your reply

Reply



Nicolas Carlo @nicoespeon · Jun 1, 2023
Previous Refa

Messages



Home



Explore



Notifications



Messages



Grok



Bookmarks



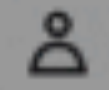
Communities



Premium



Verified Orgs



Profile



More

Post



Kelly Bodwin
@KellyBodwin

Bookmarks

@KellyBodwin

Search Bookmarks



Nicolas Carlo @nicoespeon · Apr 20, 2023

It's common for legacy code to have deeply nested statements

They make code harder to read/think about (cognitive overload) 🤔

Did you know "Guard Clauses" can help you simplify these? Now you know 🧐

Here's a JavaScript example 📄

```
function payAmount(employee, workHours) {  
  let result;  
  if (employee.isSeparated) {  
    result = { amount: 0, reasonCode: "SEP" };  
  } else {  
    if (employee.isRetired) {  
      result = { amount: 0, reasonCode: "RET" };  
    } else {  
      // logic to compute amount  
      payroll.preparePayment(employee.id);  
      employee.startDate = payroll.startDate;  
      const bonuses = employee.bonus(workHours) + payroll.benefit(employee);  
      log.addBonuses(bonuses, employee);  
      result = computeRegularPayAmount(employee, bonuses);  
    }  
  }  
  return result;  
}
```

0:48

26

99

634

72K



Nick Strayer @NicholasStrayer · Jul 28, 2022

Super excited to share with everyone the two ways I've been developing to make building your shiny app's UI faster, easier, and more pleasant: {gridlayout} and {shinyuieditor} (1/5) #RStudioConf #RStats #rstudioconf2022

gridlayout

shinyuieditor



Post



Nicolas Carlo
@nicoespeon

Follow

It's common for legacy code to have deeply nested statements

They make code harder to read/think about (cognitive overload) 🤔

Did you know "Guard Clauses" can help you simplify these? Now you know 🧐

Here's a JavaScript example 📄

```
function payAmount(employee, workHours) {  
  let result;  
  if (employee.isSeparated) {  
    result = { amount: 0, reasonCode: "SEP" };  
  } else {  
    if (employee.isRetired) {  
      result = { amount: 0, reasonCode: "RET" };  
    } else {  
      // logic to compute amount  
      payroll.preparePayment(employee.id);  
      employee.startDate = payroll.startDate;  
      const bonuses = employee.bonus(workHours) + payroll.benefit(employee);  
      log.addBonuses(bonuses, employee);  
      result = computeRegularPayAmount(employee, bonuses);  
    }  
  }  
  return result;  
}
```

0:48

4:15 AM · Apr 20, 2023 · 72.4K Views

26

99

634

123



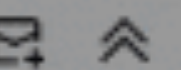
Post your reply

Reply



Nicolas Carlo @nicoespeon · Jun 1, 2023
Previous Refa

Messages





Colin 🙌🏻🌱🏃🏻 @ColinFay · May 24, 2019

Cool intro to Git:

"A GitHub Guide For People Who Don't Understand GitHub" by Lindsey
[#DEVcommunity](#)



dev.to

A GitHub Guide For People Who Don't Understand Gi
The very basics of Git and GitHub, explained



↻ 60

❤ 259



vicki @vboykis · Jun 4, 2019

Replying to [@tdhopper](#)

1. Station Eleven
2. My Sister, the Serial Killer
3. Shards of Honor
4. Ghost in the Wires
5. Bear Town
6. The Long Way to a Small, Angry Planet
7. Billy Lynn's Long Halftime Walk

💬 3



❤ 10



Zhi Yang, PhD @zhiiyang · Jun 3, 2019

Replying to [@jimhester_](#) [@hadleywickham](#) and [@ClausWilke](#)

I am pretty sure he doesn't have my mom's email address. So, here we go.

