# QUIZ

Top of Form

Question 1:

**The eq matcher checks for**

​only value

​only type

​value and type

​object identity

Answer: **only value**

Bottom of Form

Top of Form

Question 2:

**The eql matcher checks for**

​only value

​only type

​value and type

​object identity

Answer: **value and type**

Bottom of Form

Top of Form

Question 3:

**The equal matcher checks for**

​only value

​only type

​value and type

​object identity

Answer: object identity

Answer: **value** Answer: **value** Answer: **value** Bottom of Form Answer: **value** Answer: **value** Answer: **value** Bottom of Form

Top of Form

Question 4:

**The be matcher is the same as which matcher?**

​eq

​eql

​equal

Answer: equal

Bottom of Form

Bottom of Form

Top of Form

Question 5:

**Given the code below, which matchers need to be provided to result in a passing spec?**

1. **RSpec.describe 'fun with hashes' do**
2. **let(:my\_hash) { { a: 1, b: 2} }**
3. **let(:your\_hash) { { a: 1, b: 2 }}**
4. **let(:some\_hash) { my\_hash }**
6. **it 'is a test of your knowledge' do**
7. **expect(my\_hash).to \_\_\_\_\_(your\_hash)**
8. **expect(my\_hash).to \_\_\_\_\_(some\_hash)**
9. **end**
10. **end**

​**equal**and**eql**

​**equal** and **equal**

​**eq** and **equal**

Answer: **eq** and **equal**

Bottom of Form

Top of Form

Question 6:

**Given the code below, which matchers need to be provided to result in a passing spec?**

1. **RSpec.describe 'comparison matchers' do**
2. **let(:num) { 10 }**
4. **it 'is a test of your knowledge' do**
5. **expect(num).to be \_\_\_\_ 10**
6. **expect(num).to be \_\_\_\_ 15**
7. **expect(num).to be \_\_\_\_ 11**
8. **expect(num).to be \_\_\_\_ 9**
9. **end**
10. **end**

​**>=** and **>=** and **<** and **>=**

​**<=** and **<** and **>** and **>**

​**>=**and **<** and **<=** and **>**

​**>=** and **<=** and **<** and **<**

Bottom of Form

Answer: >= and < and <= and >

Bottom of FormBottom of Form