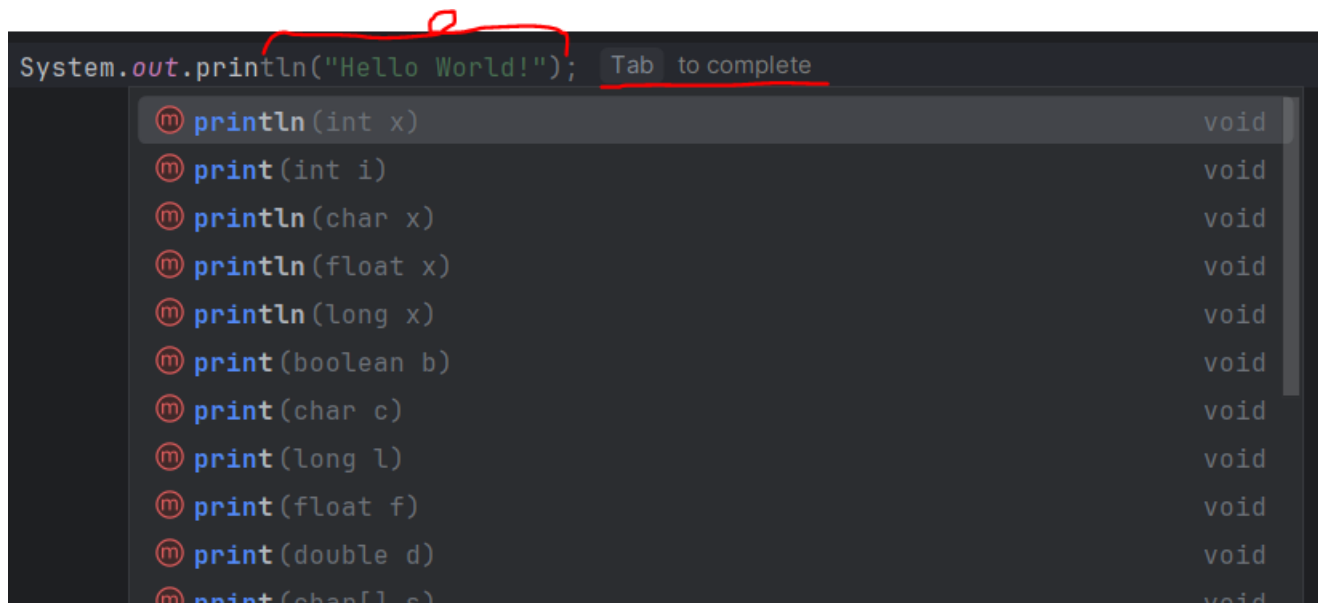


Wichtige Shortcuts:

1. Code automatisch formatieren (z.B. richtig Einrücken): **Strg+Alt+L**
2. Code kopieren: **Strg+C**
3. Mit **Strg+C** kopierten Code woanders einfügen: **Strg+V**
4. Letzte Aktion rückgängig machen: **Strg+Z**
5. IntelliJ Autovervollständigungsvorschläge annehmen (siehe Bild): **Tab**



Aufgabe 1: Variablen und Ausgaben

Man kann mit `System.out.println(einString)` den Wert von `einString` in der Konsole (unten in der Mitte bei der Ausführung des Codes) ausgeben. Folgender code:

```
1 System.out.println("PSE-Zeit ist Grillzeit")
```

Gibt folgendes aus:

```
PSE-Zeit ist Grillzeit
```

1. Erstelle eine Variable mit beliebigem Wert und gib sie aus, ändere den Wert und gib dieselbe Variable nochmal aus.
2. Erstelle zwei `int` Variablen und gib die Summe und das Produkt (die beiden Zahlen multipliziert) aus.
3. Erstelle eine Variable mit deinem Namen und schreibe genau ein Print, das dich mit deinem Namen begrüßt.
4. Kopiert folgenden Grill und nutzt prints um ihn in der Konsole auszugeben.

```
todo: hier ascii grill bitti ich kann kein latex is im tex file  
auskommentiert
```

Aufgabe 2: Scanner

Man kann während das Program ausgeführt wird auch Werte aus der Konsole einlesen. Das geht mit einem Scanner Objekt. Mit `scannerObjekt.next()` wird die nächste eingegebene Zeile eingelesen. Mit **Enter** beendet man die Eingabe.

```
1 Scanner derScanner = new Scanner(System.in);
2 System.out.println("Gib deinen Namen ein:");
3 String deinName = derScanner.next();
4 System.out.println("Du heißt: " + deinName);
```

1. Programmier einen addierer, der zwei Zahlen einliest und die Summe ausgibt. Etwas funktioniert nicht, überlegt mal warum (Auflösung in der Fußnote ¹)?
2. Programmier ein Modulo Trainer, bei dem man zwei Zahlen eingibt, dann das Ergebnis von `zahl1 % zahl2` eingeben soll und dann am Ende das korrekte Ergebnis ausgegeben wird.

¹`next()` gibt einen String zurück, wenn man die beiden addiert werden die beiden Zahlen stumpf hintereinander geschrieben, statt gescheit addiert zu werden. Dafür besitzt Scanner die `nextInt()` Funktion.

HIGHERPERFORMER-Aufgabe: Zufallszahlen

Java stellt einen Zufallszahlengenerator zur Verfügung. Dieser wird ähnlich wie Scanner benutzt und besitzt unter anderem folgende Funktionen:

```
1 Random rndGenerator = new Random();
2
3 //gibt ein zufälligen float zwischen 0.0 und 1.0 zurück:
4 float zufallsFloat = rndGenerator.nextFloat();
5
6 //gibt ein zufälligen int zwischen 0 und 9 zurück (10 wird als Obergrenze ausgeschlossen)
7 int zufallsInt = rndGenerator.nextInt(10);
```

1. Modifiziere deinen Modulotrainer sodass man die beiden Zahlen nicht mehr selbst eingibt (war auch wirklich zu leicht so), sondern so, dass diese zufällig generiert werden.
2. In folgendem code bekommt ihr einen sogenannten Seed (5-Stellig) vom eingebauten Java Zufallsgenerator. Generiert mit den mathematischen Operatoren die ihr jetzt kennt anhand dieses Seeds eine neue 10-Stellige Zufallszahl. Wir (die PSE-Vorkurs Hauptorgas) haben ein Testerprogramm geschrieben, dass versucht eure Lösung zu bewerten und eventuelle Schwächen eures Ansatzes aufzuzeigen, doch vielleicht kriegt ihr es auch hin den Test auszutricksen.

```
1 public int zufallsZahl(int seed) {
2     //seed ist die ursprüngliche 5-stellige Zahl mit der ihr rechnen könnt
3     int rueckgabeZahl;
4
5     // Schreibt euren code, der am ende eure neue zehnstellige Zahl
6     // der Variable rueckgabeZahl zuweisen soll zwischen hier...
7
8
9     // ... und hier
10    return rueckgabeZahl
11 }
```