

Betradar Unified Odds

US odds conversion in:

C#

PHP

JavaScript

Java

Python

This document contains our different code snippets for converting to US odds formats in C#, PHP, Python, JavaScript and Java.

Table of content

Local usage/console.....	1
Value ladder	1
C#	2
PHP	3
JavaScript	4
Java.....	5
Python.....	6

Local usage/console

In console (for all languages):

- Q to quit program
- | to list ladders
- Decimal numbers to convert from EU odds to US odds

Examples:

Please enter decimal value, | to display the ladders, q to quit

2.55

EU 2.55 => US 155.00000 => Rounded US 155

2.1235

EU 2.1235 => US 112.35000 => Rounded US 112

Value ladder

Together with each snippet/library, we have provided a ladder with values/odds that can be modified to your needs. The code picks the closest odds/value that is listed in the ladder.

C#

```

    * Convert EU odds to US odds.
    *
    * if EU odds > 2 then return (EU odds -1) * 100
    * if EU odds < 2 then return (-100)/(EU odds -1)
    */
private static decimal? ConvertEuOddsToUs(decimal oddsEu)
{
    decimal? oddUs;
    if (oddsEu == 1)
    {
        oddUs = null;
    }
    else if (oddsEu >= 2)
    {
        oddUs = (oddsEu - 1) * 100;
    }
    else
    {
        oddUs = (-100) / (oddsEu - 1);
    }

    return oddUs;
}

/**
 * Round the US odds to the closest value in ladder
 *
 * Ladder is a list of ordered US odds values.
 */
private static int RoundUSOdds(decimal oddsUs, List<int> ladder)
{
    var min = Int32.MinValue;
    var max = Int32.MaxValue;
    ladder.ForEach(l =>
    {
        if (oddsUs >= l)
        {
            min = l;
        }

        if (oddsUs <= l && l < max)
        {
            max = l;
        }
    });

    return Math.Abs(oddsUs) - Math.Abs(min) <= Math.Abs(max) -
Math.Abs(oddsUs) ? min : max;
}

```

PHP

```

* Convert EU odds to US odds.
*
* if EU odds > 2 then return (EU odds -1) * 100
* if EU odds < 2 then return (-100)/(EU odds -1)
*
* @param float|int $oddsEu odds in EU format
* @return float|int
*/
function convertEuOddsToUs($oddsEu)
{
    if ($oddsEu >= 2) {
        return ($oddsEu - 1) * 100;
    } else {
        return -100 / ($oddsEu - 1);
    }
}

/**
* Round the US odds to the closest value in ladder
*
* @param float|int $oddsUs odds in US format
* @param array $ladder list of ordered US odds values
* @return int rounded US odds
*/
function roundUSOdds($oddsUs, $ladder)
{
    $maxLadder = PHP_INT_MAX;
    $minLadder = PHP_INT_MIN;

    foreach ($ladder as $ladderValue) {
        if ($oddsUs >= $ladderValue) {
            $minLadder = $ladderValue;
        }
        if ($oddsUs <= $ladderValue && $ladderValue < $maxLadder) {
            $maxLadder = $ladderValue;
        }
    }
    return abs($oddsUs) - abs($minLadder) <= abs($maxLadder) - abs($oddsUs) ?
    $minLadder : $maxLadder;
}

```

JavaScript

```

/**
 * Convert decimal EU odds to decimal US odds
 *
 * if EU_odds >= 2 then US_odds=(EU_odds - 1) * 100
 * if EU_odds <2 then US_odds=(-100)/(EU_odds-1)
 */
const convertEuOddsToUs = (odds_eu) => {
  if (odds_eu === 1) {
    return null;
  } else if (odds_eu >= 2) {
    return (odds_eu - 1) * 100;
  } else {
    return -100 / (odds_eu - 1);
  }
};

/**
 * Round oddsUs to closest value from ladder.
 *
 * @param odds_us decimal US odds value not null
 * @param ladder list of ordered US odds values not null
 */
const roundUSOdds = (odds_us, ladder) => {
  let max_ladder = Number.MAX_SAFE_INTEGER;
  let min_ladder = Number.MIN_SAFE_INTEGER;

  ladder.forEach(l => {
    if (odds_us >= l) {
      min_ladder = l;
    }
    if (odds_us <= l && l < max_ladder) {
      max_ladder = l;
    }
  });

  return Math.abs(odds_us) - Math.abs(min_ladder) <= Math.abs(max_ladder)
    ? min_ladder : max_ladder;
};

```

Java

```

* Convert decimal EU odds to decimal US odds
*
* if EU_odds >= 2 then US_odds=(EU_odds - 1) * 100
* if EU_odds <2 then US_odds=(-100)/(EU_odds-1)
* @param oddsEUDouble EU odds not null
*/
public static Optional<Double> convertEuOddsToUs(Double oddsEUDouble) {
    BigDecimal oddsEu = BigDecimal.valueOf(oddsEUDouble);
    Optional<Double> oddUs;
    if (oddsEu.doubleValue() == 1) {
        oddUs = Optional.empty();
    } else if (oddsEu.doubleValue() >= 2) {
        oddUs =
            Optional.of(
                oddsEu
                    .subtract(BigDecimal.valueOf(1d))
                    .multiply(BigDecimal.valueOf(100))
                    .doubleValue());
    } else {
        oddUs =
            Optional.of(
                (BigDecimal.valueOf(-100d))
                    .divide(oddsEu.subtract(BigDecimal.valueOf(1)), MathCon-
text.DECIMAL128)
                    .doubleValue());
    }

    return oddUs;
}

/**
* Round oddsUs to closest value from ladder.
*
* @param oddsUs decimal US odds value not null
* @param ladder list of ordered US odds values not null
*/
public static int roundUSOdds(Double oddsUs, List<Integer> ladder) {
    Integer min = Integer.MIN_VALUE;
    Integer max = Integer.MAX_VALUE;
    ladder.sort(Integer::compareTo);
    for (Integer l : ladder) {
        if (oddsUs >= l) {
            min = l;
        }

        if (oddsUs <= l && l < max) {
            max = l;
        }
    }
    return Math.abs(oddsUs - min) <= Math.abs(max - oddsUs) ? min : max;
}
..

```

Python

```
"""
    Convert EU odds to US odds.

    if EU odds > 2 then return (EU odds -1) * 100
    if EU odds < 2 then return (-100)/(EU odds -1)
"""
def convertEuOddsToUs(odds_eu):
    if odds_eu == 1:
        return None
    elif odds_eu >= 2:
        return (odds_eu - 1) * 100
    else:
        return (-100) / (odds_eu - 1)

"""
    Round the US odds to the closest value in ladder

    Ladder is a list of ordered US odds.
"""
def roundUSOdds(odds_us, ladder):
    max_ladder = sys.maxsize
    min_ladder = -max_ladder - 1

    for l in ladder:
        if odds_us >= l:
            min_ladder = l
        if odds_us <= l < max_ladder:
            max_ladder = l

    if abs(odds_us) - abs(min_ladder) <= abs(max_ladder) - abs(odds_us):
        return min_ladder
    else:
        return max_ladder
```